Space-Time Interpolation Techniques

Von der Carl-Friedrich-Gauß Fakultät

Technische Universität Carola-Wilhelmina zu Braunschweig

zur Erlangung des Grades

Doktor Ingenieur (Dr.-Ing.)

genehmigte

Dissertation

von Timo Stich

geboren in Miltenberg am Main

am 3. August 1978

Eingereicht am: 15. Dezember 2008

Disputation am: 20. April 2009

Referent: Prof. Dr.-Ing. Marcus Magnor

Koreferent: Prof. Dr. Ir. Philip Dutré

(2008)

Abstract

The photo-realistic modeling and animation of complex scenes in 3D requires a lot of work and skill of artists even with modern acquisition techniques. This is especially true if the rendering should additionally be performed in real-time. In this thesis we follow another direction in computer graphics to generate photo-realistic results based on recorded video sequences of one or multiple cameras. We propose several methods to handle scenes showing natural phenomena and also multi-view footage of general complex 3D scenes. In contrast to other approaches, we make use of relaxed geometric constraints and focus especially on image properties important to create perceptually plausible in-between images. The results are novel photo-realistic video sequences rendered in real-time allowing for interactive manipulation or to interactively explore novel view and time points.

Kurzfassung

Das Modellieren und die Animation von 3D Szenen in fotorealistischer Qualität ist sehr arbeitsaufwändig, auch wenn moderne Verfahren benutzt werden. Wenn die Bilder in Echtzeit berechnet werden sollen ist diese Aufgabe um so schwieriger zu lösen. In dieser Dissertation verfolgen wir einen alternativen Ansatz der Computergrafik, um neue photorealistische Ergebnisse aus einer oder mehreren aufgenommenen Videosequenzen zu gewinnen. Es werden mehrere Methoden entwickelt die für natürlicher Phänomene und für generelle Szenen einsetzbar sind. Im Unterschied zu anderen Verfahren nutzen wir abgeschwächte geometrische Einschränkungen und berechnen eine genaue Lösung nur dort wo sie wichtig für die menschliche Wahrnehmung ist. Die Ergebnisse sind neue fotorealistische Videosequenzen, die in Echtzeit berechnet und interaktiv manipuliert, oder in denen neue Blick- und Zeitpunkte der Szenen frei erkundet werden können.

Zusammenfassung

Heutzutage sind die Ergebnisse fotorealistischer Bildberechnungen von dynamischen und komplexen Szenen täglich auf Kinoleinwänden und im Fernsehen zu sehen. Das Modellieren und die Animation solcher fotorealistischer Szenen ist jedoch sehr arbeitsaufwändig und die Qualität nicht zuletzt Abhängig von den Fähigkeiten der 3D-Artists. Insbesondere dann, wenn die Bilder in Echtzeit berechnet werden sollen, wie dies im Fall von Computerspielen notwendig ist, ist diese Aufgabe um so schwieriger zu lösen.

Anstatt Szenen so genau wie möglich im Computer 3-Dimensional abzubilden und diese dann wieder durch Berechnungen in 2-Dimensionale Bilder umzuwandeln, bietet es sich alternativ an, mehrere aufgenommene Bilder zu kombinieren um ein gewünschtes Ergebnis zu erzielen. Allerdings beruhen auch solche Verfahren häufig auf der Rekonstruktion von 3-Dimensionaler Geometrie, was zu Einschränkungen in der Aufnahmemodalität, des Kameraufbaus und der Szene selbst führt.

Die in dieser Dissertation beschriebenen Verfahren umgehen diese Einschränkungen und zeigen, wie die Information aus den Bildern alleine genügt um plausible Ergebnisse zu erzielen. Diese sind nicht notwendigerweise physikalisch korrekt im strikten Sinne, werden aber als fotorealistisch vom menschlichen Betrachter wahrgenommen. Um hierfür neue Verfahren und Algorithmen zu entwickeln, nutzen wir abgeschwächte geometrische Einschränkungen der Lösung und berechnen eine genaue Lösung nur in den Bildbereichen, die wichtig für die menschliche Wahrnehmung sind.

Zusammenfassend befasst sich diese Arbeit mit neue Verfahren zur Erzeugung von Videosequenzen aus einer oder mehreren Aufnahmen in Echtzeit. Der erste Teil beschäftigt sich mit der Erzeugung neuer Videosequenzen natürlicher Phänomene (z.B. Feuer) basierend auf ihrer quasi-periodischen

Natur. Dann behandeln wir generelle Aufnahmen mit mehreren Kameras. Wir führen Verfahren ein, die plausible Interpolationsergebnisse von Bildern, die verschiedene Blick- und Zeitpunkte zeigen, ermöglichen. Mit unserer Methode zur Schätzung des Zeitversatzes zwischen unsynchronisierten Aufnahmen und deren Einbettung in einen passenden Navigationsraum erreichen wir Raumzeitinterpolation von unsynchronisierten und unkalibrierten Aufnahmen mehrerer Kameras. Besonders die Möglichkeit diese Effekte mit Aufnahmen die mit Standardkamers gemacht wurden zu erzielen, hilft die Kosten zu reduzieren und bildet eine Brücke zwischen Laborexperimenten und der realen Filmproduktion.

Acknowledgements

Many people supported and inspired me during the work on my thesis. First and foremost I am grateful to my supervisor Prof. Marcus Magnor. I enjoyed having the opportunity to work both at the Max-Planck-Insitute as well as at the TU Braunschweig together with you. You have shown me interesting new research directions, gave me the freedom to pusue my own ideas and motivated me for the major conference deadlines. I am also deeply grateful for the many conferences I was able to visit during that time.

I would especially like to thank all my colleagues that have worked with me on previous publications, in particular Georgia Albuquerque, Douglas Cunningham, Christian Linz, Christian Lipski, Benjamin Meyer and Christian Wallraven. It has been both very fruitful and a great pleasure working with these splendid researchers. Thanks to all members of the Graphics-Optics-Vision Group in Saarbrücken and the Computer Graphics Lab in Braunschweig for the discussions, help and for making it such great environments to work at. Thank you, Anja, for making the admisitrative part of our work as easy as possible! Special thanks to Anita, Christian, Ivo, Kristian, Nicole and Martin for proof-reading drafts of this dissertation.

I also like to thank all the people who participated in the various video recordings for the projects, both as actors and as support. In particular the Capoeira, Frisbee and the Kobudo university sport groups, the dancers Yuki and Mona as well as Prof. Wand for performing as fire breather and Ulli Becker and Peter Dargel for providing the recording locations. Special thanks is due to Andreas who worked as a research assistant relentlessly with me on recording and processing all the video data to make the deadlines.

I am most grateful to my parents Frank and Claudia. You have always supported me and spawned my interest in computers and science. Nicole,

thank you for your encouragement, love and motivation - thanks for always being there for me!

Contents

1 Introduction **1**

1.1 Main Contributions 2

1.2 Thesis Overview 3

2 Background **5**

2.1 The Plenoptic Function 5

2.2 Human Vision and Connections to Computer Vision 6

2.3 Image Morphing 12

2.3.1 Spatial Transformations 13

2.3.2 Image Blending 19

3 Related Work **21**

3.1 Natural Phenomena 21

3.2 View and Time Interpolation 24

3.3 General Image Interpolation 29

3.4 Video Synchronization 30

3.5 Perceptual Adaptive Graphics 31

4 A Dynamic Image Space Model for Flames **33**

4.1 Introduction 33

4.2 Flame Appearance 34

4.2.1 Flame Shape and Texture 35

4.2.2 Estimating Shape Parameters 35

4.2.3 Estimating Texture Parameters 37

4.3 Flame Dynamics 38

4.4 Rendering 39

4.5 Results 40
4.5.1 Control and Interaction 41
4.6 Summary 42

5 Key-frame Animation of Natural Phenomena from Video Sequences 43
5.1 Introduction 43
5.2 Video Analysis 43
5.2.1 Video Trajectories 44
5.2.2 Video Blocks 46
5.3 Video Synthesis 46
5.3.1 Sequencing 46
5.3.2 In-between Images 47
5.4 Results 48
5.5 Summary 48

6 Image Morphing for Space-Time Interpolation 51
6.1 Introduction 51
6.2 Improving Feature-Based Warping 52
6.2.1 Per-Feature Optimal Weighting Parameters 53
6.2.2 Per-Pixel Warp Field Correction 54
6.3 Perception-motivated Non-linear Blending 56
6.3.1 Classifying Image Differences 56
6.3.2 Non-linear Image Blending 57
6.4 Plausible Feature Animation 58
6.5 Motion Layers 59
6.6 Implementation 59
6.7 Results 60
6.8 Summary 60

7 Automatic Perception-Aware Space-Time Image Interpolation 63
7.1 Introduction 63
7.2 A Novel Image Deformation Model for Time and View Interpolation . . 64
7.3 Estimating the Image Deformation 65
7.3.1 Matching of Edge Pixels 66

7.3.2 Estimating the Local Homographies . . . 69
7.3.3 Translet Optimization . . . 69
7.3.4 Per-Pixel Correspondences . . . 70
7.3.5 Multiple Iterations and User Interaction . . . 71
7.4 Interpolation Rendering . . . 73
7.4.1 Warping with Occlusions . . . 73
7.4.2 Feathering . . . 74
7.4.3 Multiple Image Interpolation . . . 75
7.5 Results . . . 75
7.6 Summary . . . 76

8 A Psychophysical User-Study on Image Interpolation 81
8.1 Introduction . . . 81
8.2 Perceptual Criteria for Image Interpolation . . . 81
8.3 User Study . . . 82
8.3.1 Stimuli . . . 82
8.3.2 Experimental design . . . 84
8.3.3 Analysis . . . 86
8.4 Summary . . . 87

9 Estimating Time Difference of Uncalibrated and Non-Stationary Cameras 93
9.1 Introduction . . . 93
9.2 Problem Formulation . . . 94
9.3 Frame-accurate Temporal Alignment . . . 94
9.4 Achieving sub-frame accuracy . . . 97
9.5 Results . . . 99
9.6 Summary . . . 102

10 Multi-View and Time Interpolation in Image Space 105
10.1 Introduction . . . 105
10.2 Navigation Space . . . 106
10.2.1 Axis Definition . . . 107
10.2.2 Tetrahedralization . . . 109

10.3 Rendering 109
10.4 Summary 110

11 Discussion and Conclusions **113**
11.1 Summary 113
11.2 Conclusions 115
11.3 Future Work 116

Bibliography **117**

1

Introduction

Photo-realistic renderings of dynamic and complex scenes are screened in cinema and seen on TV every day. While computer generated footage is most common in the form of special effects, even rendered full featured films such as Final Fantasy: The Spirits Within (2001) and more recently Beowulf (2007) have been produced. The modeling and animation of photo-realistic scenes and movies however still requires a lot of work and skill of artists. This still holds even if motion tracking, 3D scanners and reflectance field acquisition devices are used to capture the properties of real objects and actors to be reproduced in virtual environments. Rendering images in real-time on commodity hardware for computer games is even more demanding. The limited number of processable triangles and shader computations per frame make it necessary to employ clever tricks, bending and simplifying the physical reality to create plausible realities.

Instead of modeling scenes as accurate as possible in 3D and using rendering techniques to again produce 2D images, another approach is to make use of recorded footage directly, since those are by definition photo-realistic. The task is then to manipulate and combine these photos and videos of real-world scenes in such ways that they remain photo-realistic but show the scene as intended by the artist or director. However, also in the image based approaches most works rely on the reconstruction of 3D geometry which poses restrictions on the acquisition modalities such as the cameras in use, their setup and the scene itself.

In this thesis, the goal is to address these limitations and to show how the information present in the images alone can be used to create plausible results. These might not

be real in the strict physical sense, but do appear indistinguishable from the physical reality to human observers. To achieve this, we make use of relaxed geometric constraints and focus on motion perception properties of the human visual system to find new algorithms and approaches to create photo-realistic video sequences from recorded footage suitable for real-time rendering.

1.1 Main Contributions

Figure 1.1: Results rendered with the algorithms and methods presented in this thesis. These image composites of interpolated images show examples of multi-exposure (left) and motion distortion (middle, right) effects.

Throughout the course of this dissertation, parts have already been presented at various conferences and published in conference proceedings and journals [87, 136, 137, 138, 139, 140, 141]. These publications are the foundation of this thesis which incorporates them under the framework of single and multi-video image interpolation and presents improvements and updated results. The main contributions are:

- The development of a learnable appearance and motion model of flames, Chapter 4. The space spanned by the model and learned from recorded video sequences is the basis for creating interactive, endless novel image sequences of burning flames and combines the advantages of dynamic and video textures.

- A novel video texture approach for natural phenomena scenes based on video trajectory analysis and image interpolation based on mass transport, Chapter 5.

- A layer-based semi-automatic image morphing method based on a standard image morphing technique, Chapter 6. Our approach reduces the necessary user interaction and handles occlusions with a perceptually motivated non-linear blending function.

- A fully automatic image interpolation method that is capable of interpolating multi-video content in view and time, Chapter 7. The advantage of the method is the combination of perspective geometry constraints in the form of estimated local homographies while focusing on image properties important to the human visual system. This makes it possible to plausibly interpolate in view and time and handles occlusion and motion discontinuities gracefully.

- A non-intrusive sub-frame accurate time offset estimation method for non-stationary cameras, Chapter 9. This estimation makes it possible to bring recorded video sequences to a common time coordinate system without the need for synchronization of the cameras.

- A representation of multi-view camera data which allows for intuitive, independent and smooth space-time interpolation from unsynchronized and uncalibrated footage, Chapter 10.

Taken together, these components form novel approaches to space-time interpolation of single and multi-view recordings and allow creating special effects as shown in Figure 1.1.

1.2 Thesis Overview

This thesis is structured as follows: In Chapter 2 we introduce and review the concepts and terminology used throughout this thesis. Work related to our topic is discussed in Chapter 3. Then, in the Chapters 4 and 5, we focus on natural phenomena, and propose two image-based motion models to manipulate and create novel image sequences with the desired properties from recordings of a single camera. In Chapter 6 we introduce improvements to the warping method of Beier and Neely [9] to make it more suitable to space-time interpolation, introduce a non-linear blending scheme to reduce the visibility of remaining artifacts by taking properties of the human visual system into account and

discuss its applicability to view and time interpolation. Then, we take the next step in Chapter 7 and propose a novel fully automatic approach to image interpolation of unsynchronized and uncalibrated images which is based on perspective geometry constraints while focusing on human motion perception to reduce the complexity of the problem while maintaining the same visual quality. We confirm the algorithms ability to create perceptually correct imagery in a user study, Chapter 8 . With the estimation of the temporal offset of video sequences taken from uncalibrated, non-stationary cameras, Chapter 9, we introduce a space-time embedding of these recordings that gives rise to a multi-image interpolation scheme. This then allows to intuitively and independently navigate in view and time without the necessity to reconstruct the 3D scene, camera parameters and thus makes free viewpoint videos much easier to capture as discussed in Chapter 10. Finally, we conclude this thesis in Chapter 11 and give an outlook into future research enabled by the research so far.

2

Background

2.1 The Plenoptic Function

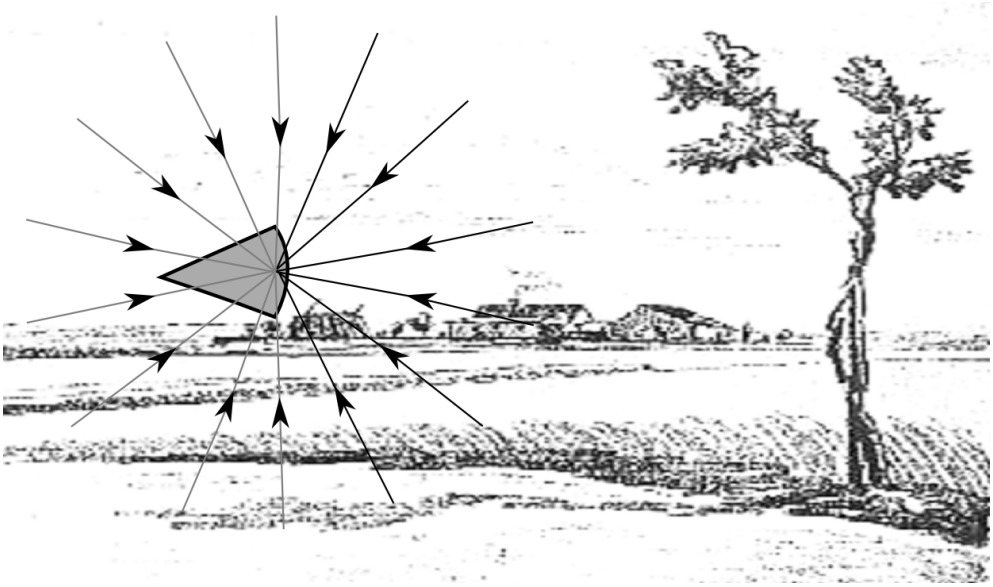


Figure 2.1: The plenoptic function describes all available information to an observer for a given position and point in time. Depicted here are the radiance samples recorded by a camera resulting in an image of the scene.

The world is made of 3D objects. The eye of the observer is a sensor to measure these objects. However, the objects do not directly communicate their properties to the observer. Rather, they fill the space surrounding them with patterns of light rays.

These patterns are described by the *plenoptic function* [1]:

$$P(\phi, \rho, \lambda, t, O_x, O_y, O_z) \tag{2.1}$$

where ϕ, ρ describe the viewing angles, λ denotes a wavelength, t is a point in time and O_x, O_y, O_z denotes the position of the observer (cf. Figure 2.1). The plenoptic function describes the light of any wavelength that can be observed at any time and point in space from any direction and wavelength. Thus the communication by visual means between the surrounding world and any observer of it is fully described by this function. Clearly, only a small portion of this 7-dimensional function can be measured by any sensor. For example, images taken with a camera as well as the retinal image on the eye are just samples of the plenoptic function for fixed O_x, O_y, O_z, t [1].

In this thesis, we will focus on methods based on samples of the plenoptic function such as images and video sequences. Our goal is to reconstruct parts of the plenoptic functions from recorded samples. However, we do not aim at reconstructing the correct nor the full plenoptic function as this is impossible due to the complexity, dimensionality and the relatively high cost of sampling it. Instead, we focus on reconstructing a partial plenoptic function, that when taking novel samples, will result in consistent and convincing new image sequences. Thus, we define quality in terms of how the human visual system processes the result of our reconstruction without noticing visual artifacts rather than how close the reconstruction is to the physically correct function.

2.2 Human Vision and Connections to Computer Vision

It has been proposed that the basic task of early human vision is to extract as much information as possible about the structure of the plenoptic function [1]. Of special interest are significant local changes in the plenoptic function and their orientation. In an arithmetic sense these local changes are equivalent to derivatives of the plenoptic function. Thus measuring low order directional derivatives fits well the goal of efficiently capturing the structure of the plenoptic function. Interestingly, the stencil of such derivative filters as depicted in Figure 2.2 resembles the layout of some cell agglomerates on the retina and the LGN (Laterate Genicule Nucleus), e.g. [56]. They consist of

[1] For the sake of simplicity, the fact that such images are the results of integrals of the plenoptic function over the exposure time and the space angle of the receptors is neglected throughout this thesis.

regions of different cell types which are connected to form a single output. Since the regions are behaving differently when the are exposed to light, the first ones are excited the second ones are inhibited, they react only to very specific light intensity patterns.

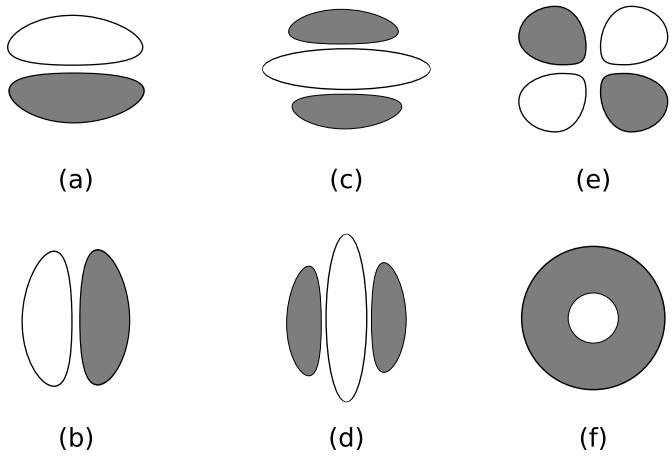


Figure 2.2: Two-dimensional receptive fields resembling similarity to low-order derivative operators on the plenoptic function. Dark and light areas illustrate regions of different cells. The first are depolarized by light, while the second are polarized by it. (a-b) can be interpreted as first order differences, (c-e) are second order differences and (f) is the isotropic Laplacian ∇^2.

Since the experiments by Hubel and Wiesel [57], it is known that these cell agglomerates, the so called simple and complex cells, in the visual cortex are sensitive to motion in a specific direction. They are layed out in cell columns in the brain where each column corresponds to a specific range of directions. This means, an intensity edge moving in one direction while result in a measurable peak in the cells most excited by the motion and no change in the others. This specific peak thus on the other hand allows inferring the movement direction of the seen motion.

At first cells that are sensitive to specific motion directions have been discovered. But if the goal would be to take measures of the plenoptic function one would also need to measure the spatial variations. Research in this direction however has shown that humans are actually much more sensitive to moving edges than to stationary ones [56]. Further, the retina is highly specialized as it has only a very small region, the *Fovea*, where the density of photo-receptor cells is high enough for fine-detail vision, e.g. [99]. To compensate for these limitations, the eye is constantly in motion, the so called

saccades. If a human subject observes for example a static image, the retina is fixed on a specific point of interest for a short period of time and then abruptly moves to another point of interest. This is especially noticeable during reading and was first reported by Louis Emile Javal in the late 19th century.

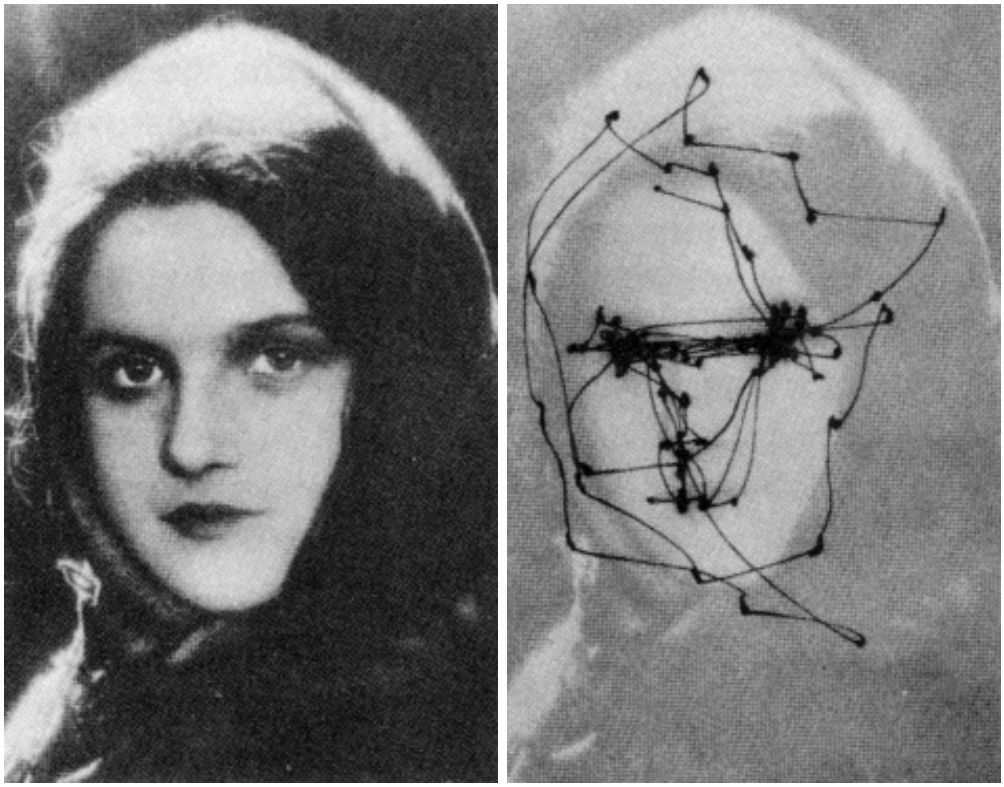


Figure 2.3: The motion of the eye recorded by Alfred Yarbus using his eye tracker. During observing the image, the eyes alternately come to rest for a short time and then abruptly jump to a new location, the so called *saccades*. (Images taken from [166])

Later Yarbus published his influential work [166] on saccades where he also showed that the saccades are task dependent, such as counting or remembering certain features, using an eye tracker as depicted in Figure 2.3. However, even if the gazing direction of the eye of a human subject is fixed on a certain point there is still some motion left. Contrary, if a static stimulus is absolutely fixed relative to the gazing direction of the eye it vanishes from the subjects perception [33, 115]. Only due to these very small motions in the order of 1 to 2 arc minutes, the so called *microsaccades*, static spatial edges become moving edges and thus visible. It seems that by implementing

simple space/time edge detectors in combination with saccades and microsaccades, human vision indeed measures the spatial and temporal local changes of the plenoptic function. Although human vision of the world is binocular and thus makes it possible to measure another dimension of the plenoptic function (i.e. disparity), we focus in this thesis only on monocular vision, which is the case when observing pictures and videos.

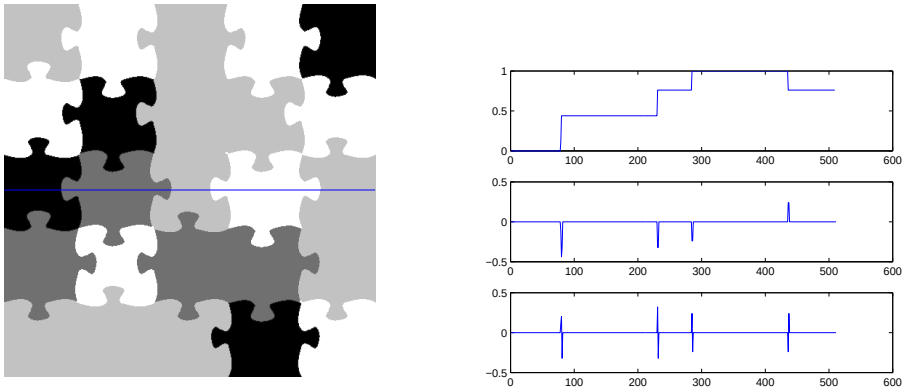


Figure 2.4: Edges are intensity changes in an image (left). (Right) shows the 1-D intensity profile of the blue region and the resulting responses of directional first and second derivate filters. Edges appear as peaks in the first order and zero-crossings in the second order derivatives.

At this point we make a connection to computer vision and image processing. As discussed, edge detection is an important mechanism in human vision and has also been well researched in the field of computer vision. Consequently, Marr and Hildreth [82] discussed the theory of edge detection and its implementation in the human vision system based on signal processing observations. Edges in an image are directional changes in intensity as illustrated in Figure 2.4. Thus an edge detector could measure these changes using first and second order derivatives. Further, they also consider that changes happen on different scales. By first convolving the signal with a Gaussian kernel prior to building the derivatives, selects the scale of the frequencies of interest. Marr and Hildreth argue the best suited filter fulfilling these goals is the Laplacian of Gaussian filter, $\nabla^2 G_\sigma$, where σ defines the standard deviation of the Gaussian and thus selects the scale of interest. Other very successful edge detectors, like the Canny edge detector [21] also apply this concept of pre-smoothing the input but rather evaluate the responses of

the first order derivatives. In conclusion, the numerical derivative operators bear great similarity to the receptive fields of simple and complex cells depicted in Figure 2.2.

The mechanisms discussed so far are part of the early vision in the processing of the visual input. Overall the human visual system is organized hierarchically, where different layers of information processing output and get input from the next higher level, forming complex interaction patterns [56]. In the following we point out properties of higher level vision regarding motion and color processing related to the topics discussed in this thesis.

After the first stages of visual input processing the input on the retina is analyzed in one-dimensional measurements. This however makes the interpretation of the underlying motion reconstruction ambiguous, also known as the *aperture problem* [83]. These ambiguities can often be resolved by combining all the local motion estimates into a global motion solution. However, one could also consider local features that are unambiguous because they measure two instead of a single motion direction, such as points and corners, to resolve ambiguities. Since both approaches seem plausible, the question arises which of the two approaches, the global or the local approach can explain the properties of human motion perception capabilities. Adelson and Movshon [2, 92] researched this in several experiments. For example, while the perception of the motion direction for a stripe pattern is quite clear, an interesting observation was made when such simple motion patterns overlap. Depending on the similarity of the simple motion patterns, different motion perceptions are the result. The perceived motion, if the patterns are compatible, that is have similar spatial frequency and similar contrast, is the vector addition of motions as depicted in Figure 2.5. However, if the patterns are different, the perceived motion will be that of overlayed, transparent patterns moving in two distinct directions. While the global motion understanding model and the local model based on unambiguous features often result in the same prediction of the perceived motion one can create stimuli where the predictions differ [2]. In this case the local model that relies on unambiguous features seems to be the preferred solution. However if the view is restricted, e.g. due to bad lighting or blur, the global model takes over. Thus it seems that both models are useful in understanding motion perception.

Another aspect of motion perception on an even higher level is happening in the medial superior temporal (MST) area. Here cells respond to coherent motion patterns such as radial, circular and spiral motion [91]. In contrast to the cells in the previous

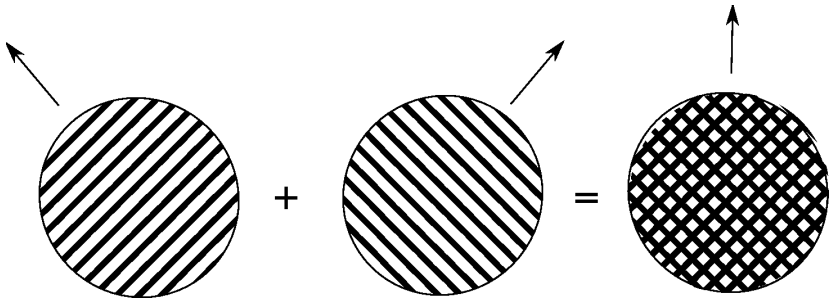


Figure 2.5: Stimulus used in the study of perceived motion by Adelson and Movshon [2, 92]. When the stimulus of two moving sets of bars at 45-degree angles is superposed the perceived motion is that of an upward motion.

layers which are sensitive to translational motion and have rather small receptive fields these cells have very large receptive areas. This processing step for robust estimation of optical flow, e.g. for ego motion, is also similar to estimating parametric motion models from low level measurements such as the ones discussed in Section 2.3.1 in computer vision.

Besides the processing of shape and motion, another dimension that the human visual system is able to measure is the wavelength of light. Since Newton's experiment in 1704 where he split up light using prisms, it is known that the perceived color is a mixture of different wavelengths [95]. Later it became clear that all colors can be described by mixing three wavelengths which was called *trichromacy*. Young proposed [167] that the human retina is made of "particles" that measure the colors red, green and violet. These particles have later been identified as the cone cells on the retina and are tuned to the wavelengths associated with these colors. However, the perceived color of a patch is not only dependent on the reflected wavelength but also on the spatial neighborhood. For example, a white patch will be interpreted as white under different lighting conditions, such as office light or sun light. This is called *color constancy* and was discovered by Edwin Land [67]. The corresponding cells that make this possible have a center-surround receptive field, where the center and surround react to the color pairs red/green and blue/yellow in opposed presence (e.g. off/on). Interestingly these color sensitive cells are indifferent to orientation in contrast to the shape/motion cells [74]. However, also shapes that are distinguishable only by color differences and not by luminance differences are still perceived. Thus colors, like red and green, is just one

means by which shapes manifest themselves.

Another important factor in human vision is the concept of grouping and segmentation as introduced by Gestalt theory [156]. Grouping is used in many instances such as the grouping of patterns, features and color regions. For example the theory of textons [62] proposes to group simple features to create meaningful agglomerates. This idea has also been transported to computer vision to facilitate image understanding by generative models [171]. A large body of work in computer science also deals with the topic of grouping similar image regions to achieve meaningful segmentations. While the semantic segmentation in regions such as foreground and background already relies on a high level of image understanding, recently other approaches have been proposed that are motivated by the early grouping processes of human vision. Locally grouping pixels of similar attributes into larger but still comparatively small agglomerates, so called superpixels [42, 114], leads in general to more robust results when used as computational atoms in comparison to single pixels. In human as in computer vision these groupings help in understanding the basic structure of the visual input which supports a higher level understanding of the surrounding world.

In this quick introduction to human vision we have only scratched the surface of the body of knowledge and theories about how human vision works. We refer the interested reader especially to the books by Hubel [56] and Marr [81] for more detailed coverage of the topic. While there are still many open questions we can already put the knowledge we have to our advantage. Especially, focusing on the specific image properties that are important to human vision, reduces the complexity of difficult tasks such as the estimation of correspondence and motion between images while keeping or even improving the quality of the obtained results as is demonstrated in this thesis.

2.3 Image Morphing

Image morphing is a technique in computer graphics that is used to create smooth transitions between pairs of images. Its first appearance dates back to the experimental art of Tom Brigham in the early 1980s and has often been used for special effects in movies [9, 160]. The quality of the image morphing result is depended on the quality of the two steps involved: image warping (deformation) of the images followed by blending of the warped images. The goal of image warping is hereby the *geometric* deformation

of the images such that features of the images are put in accordance to each other. Image blending then achieves the transition in *appearance* by per-pixel blending of the color values to produce the in-between image. Put into an equation the in-between image $I_{1,2}(\alpha)$ is defined as

$$I_{1,2}(\alpha) = B(W(I_1, \alpha), W(I_2, 1-\alpha), \alpha) \tag{2.2}$$

with $\alpha \in 0..1$ and $I_{1,2}(0) = I_1$ and $I_{1,2}(1) = I_2$, W denotes the geometric deformation, or warping function, and B the blending function. By computing a series of in-between images $I_{1,2}(\alpha)$ for regularly spaced α results in smooth transitions between image pairs.

Most image morphing techniques differ in the warping that is applied to the images rather than the blending which is typically a simple linear cross-dissolve. This reflects the fact that plausible transitions, are above all dependent on the plausible transformations of geometric features such as edges and corners. This is also in correspondence with the knowledge we have about the human visual system. As discussed in the previous section it focuses on edge measurements of the plenoptic function to understand the surrounding. In the next sections we will discuss several warping and blending techniques suitable to image morphing.

2.3.1 Spatial Transformations

In this section we introduce common spatial transformations that are used in image warping applications. A spatial transformation is a mapping between two coordinate systems that establishes correspondence between an image and its warped counterpart. This can be for example an artistic deformation of the image or a deformation that is applied to bring the image in correspondence with another image. We will focus in this section on the most basic transformations and refer the interested reader to more extensive introductions such as in the works by Wolberg [160] and Hartley and Zisserman [52].

In mathematical notation, a spatial transformation can be described either as the relation of input coordinates $\mathbf{x}_1$ to target coordinates $\mathbf{x}_2$ or vice versa:

$$\mathbf{x}_2 = W_F(\mathbf{x}_1) \tag{2.3}$$

and

$$\mathbf{x}_1 = W_B(\mathbf{x}_2) \tag{2.4}$$

where W_F, W_B are 2D mapping functions that define the transformation. Equation 2.3 is called the forward mapping, where each input position is associated with an target position and 2.4 is called the backward or inverse mapping where for each position in the target image a correspondence is defined with the input (cf. Figure 2.6).

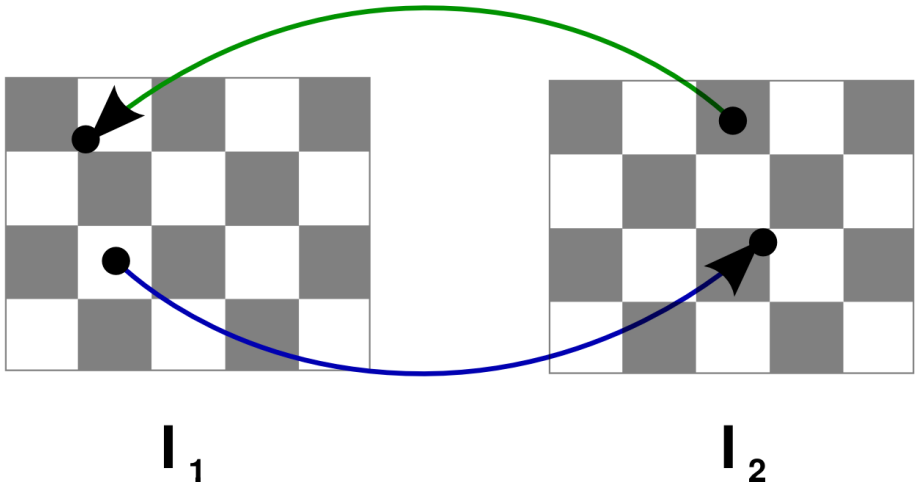


Figure 2.6: Spatial transformations on a regular lattice can be implemented in two ways. Forward warping (blue) is the mapping of input coordinates on the lattice to target coordinates not necessarily on the lattice. Backward warping (green) is the opposite direction where the inverse spatial transformation is used to map lattice target coordinates back to non-lattice input coordinates. See text for how the sampling issues can be resolved.

Since we are dealing with digital images which are defined on uniform integer lattices both approaches lead to sampling problems when the transformation is applied. In the most naive implementation of forward warping, single pixels of the input image are copied to the corresponding position in target coordinate system. Since the mapping in general maps integer positions to real valued positions, this must be properly handled to avoid artifacts such as holes and overlaps. To overcome this problem, pixels can be represented by quads and the four corners instead of the mid-points are transformed. Then the image surface is still contiguous after the mapping. However, when several quads end up in the same target pixel, the contribution of all quads needs to be accounted for. Forward warping can be implemented on modern programmable graphics hardware to run in real-time by rendering 2D meshes where one quad represents one pixel and vertex shaders are used to achieve the deformation.

With backward mapping the position of the target coordinate system is computed in the input coordinate system. To achieve this one must compute the inverse spatial transformation. While for classes of transformations, such as the ones discussed in the next section, an inverse exists and is straight forward to compute, it must not necessarily exist in the general case. However, if an inverse exists it is assured that each pixel in the target is computed and no holes or overlaps will occur. As with forward mapping the value of a target pixel is often dependent on more than one input pixel and thus filtering has to be applied to avoid aliasing artifacts. This can also be easily implemented on graphics hardware to run in real-time by rendering a quad that represents the target image and fragment shaders performing per-pixel texture lookups with an appropriate texture sampling strategy.

Projective Transformations In this section we are interested in a special class of spatial transformations, the so called projective transformations. As part of 2D projective geometry, they form a group of invertible mappings between points in $\mathbb{P}^2$ and that map lines to lines [52]. Put into an equation we can define a projective transformation as a linear transformation on homogeneous 3-vectors represented by a non-singular 3×3 matrix H:

$$\mathbf{x}_2 = H\,\mathbf{x}_1. \tag{2.5}$$

Since we are working with homogeneous coordinates, H and kH for all non-zero k describe the same transformation. To express this equality we also write

$$H \cong kH\ , \forall k \neq 0 \tag{2.6}$$

in this thesis.

These transformations form a group, because the inverse of a projective transformation is again a projective transformation as is the combination of two projective transformations. Note also, that since these transformations are invertible, the image deformations can be either implemented by forward or backward mapping as discussed in the previous section. As put forward by Klein in his "Erlangen Program" [63], one can categorize transformations by the invariants or preserved geometric properties of the transformation. In the following we will discuss subgroups of projective transformations, that are specialized versions of projective transformations and their geometric invariants. Here, subgroups again implies that the combination and the

inversion of these specialized projective transformations again is such a specialized projective transformation and so is the composition of two. A summary of the specialized transformations along with an example are listed in Table 2.1.

Group	Deformation	Geometric Invariants
Similarity 4 dof		Ratio of lengths, angles
Affine 6 dof		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines
Projective 8 dof		Concurrency, collinearity

Table 2.1: The hierarchy of projective transformations. Each group is categorized by its degrees of freedom (dof) and the geometric properties invariant under these transformations. The listed transformations form a hierarchy as each group is a subgroup of the next group, from top to bottom.

The most specialized subgroup of the projective transformations that is interesting to us are the *similarity transformations*. They can be decomposed into translation, rotation and isotropic scaling. The transformation matrix H_s describing similarity transformations is of the form

$$H_s = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ -h_{12} & h_{11} & h_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

and thus has 4 degrees of freedom. To estimate the parameters of a similarity transformation two point correspondences between two images are necessary. For example, a similarity transformation describes the rigid translation of a planar 3D object parallel to the imaging plane of a perspective camera.

The next subgroup of projective transformations are the affine transformations.

They are represented by transformation matrices of the form:

$$H_a = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{2.8}$$

and thus have 6 degrees of freedom. The similarity transformations are a subgroup of the affine transformations. Affine transformations can be understood as non-singular linear transformations followed by a translation. They can also be understood as the combination of rotations and non-isotropic scaling. Affine transformations are defined by three point correspondences between a pair of images. With large focal lengths and distant objects, such as in aerial photography, the imaging process is often well enough approximated by a so called affine camera [52]. In this context, planar objects undergoing any rigid 3D deformation can be described by affine transformations.

The most general projective transformations are described by the full 8 degrees of freedom and are of the form

$$H_p = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}. \tag{2.9}$$

They are also synonymously referred to as homographies or collinearities. Again, similarity and affine transformations are subgroups of homographies. A homography is estimated from 4 point correspondences and supports translation, rotation, shear, anisotropic scaling and perspective foreshortening. In the context of perspective imaging, the point to point relation between a planar object undergoing any rigid 3D deformation between a pair of images can be described by a homography.

Besides this single camera interpretation, homographies can also be interpreted in terms of pairs of perspective cameras [52]. The relation of corresponding points on a 3D plane between two views is defined by a homography as depicted in Figure 2.7. In the two view case and a planar scene, then not only the homography describes relations but also the more general epipolar geometry. This additional relation brings forward new possibilities, such as for example to estimate the fundamental matrix from two homographies and the reconstruction of the actual 3D plane if the camera matrices are known. For an extensive coverage of the relation to epipolar geometry, we refer the interested reader to the book of Hartley and Zisserman [52].

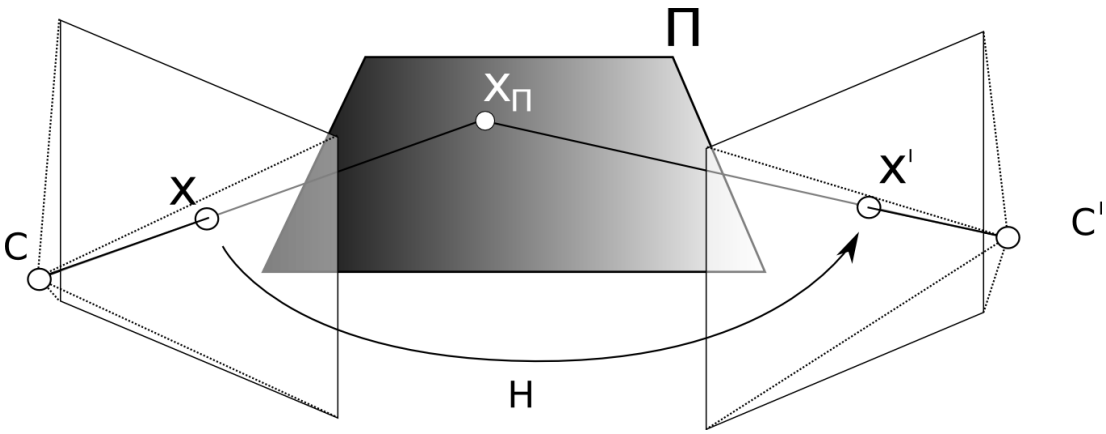


Figure 2.7: The relation between images of corresponding points on a 3D plane between two views is defined by a homography. It is a projective relation since it is defined by the intersection of rays and planes. The homography H thus maps the image of all X_π in the first view x to their corresponding image x' in the second view.

Piecewise Spatial Transformations So far we have introduced global spatial transformations that are defined on the whole image lattice. However, images of real scenes can only in very special cases be brought into correspondence by such simple transformations. One example where this is possible is the case when the recording camera was rotated around the viewpoint to take images. Then a single homography brings these into correspondence and can thus be used to create panoramic mosaics, as for example discussed by Shum and Szeliski [128]. On the other hand, one can try to estimate a transformation for each individual pixel in the images, resulting in a dense correspondence map or dense optical flow field. Then complex motions can be modelled by vector fields. However, transformations can be ambiguous for regions of the images especially when there is little local texture. To resolve this issue, additional constraints such as the heuristic to optimize smoothness of the solution are typically used, as for example in the seminal work by Horn and Schunk [55].

In-between these two extremes, complex transformations can be modeled as combinations of simple transformations with local influence. The flexibility is thus dependent on the number of simple transformations. Their influence can be chosen appropriately to get control of the robustness and flexibility of the resulting motion, cf. Figure 2.8. The first direction that one can take is to segment the images into disjoint regions, so called layers, that are then separately transformed [154]. Thus the difference to the

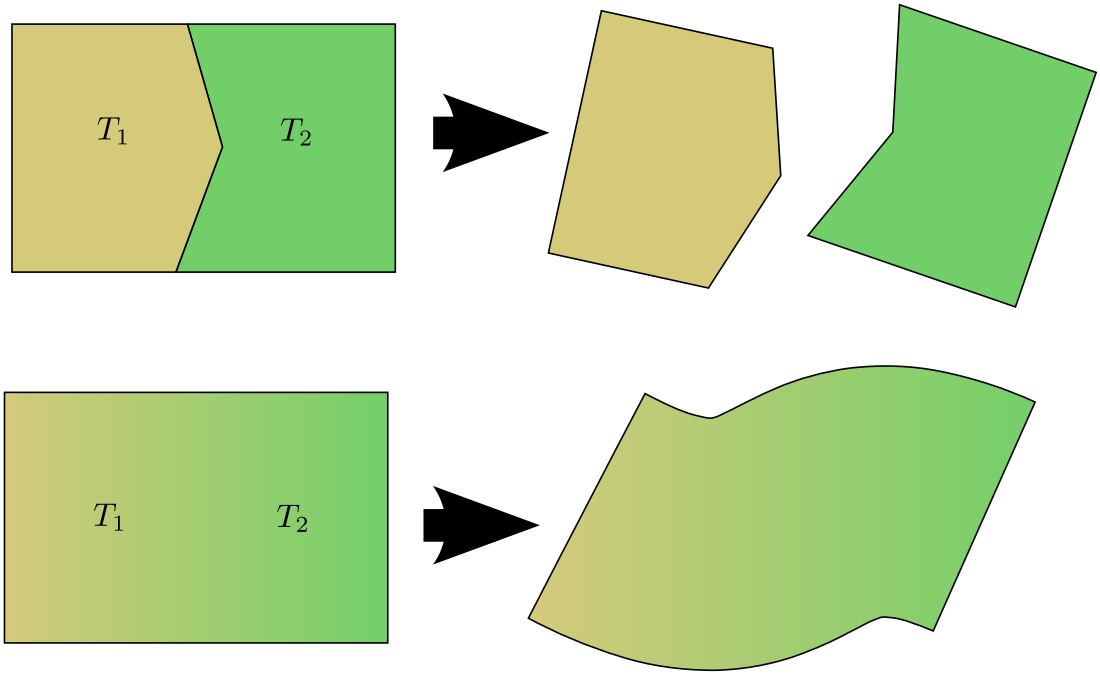


Figure 2.8: Complex transformations can be created by composites of simple transformations. This figure shows two examples of different approaches to achieving this. In the first row each transformation has a specific influence and the resulting transformation is discontinuous. The second row shows a smooth mixing of the transformations based on distance to the transformations' fix-points.

global transformations is that we define a set of transformations T_i that are defined on disjoint regions.

The second direction we are going to consider is to interpolate between different spatial transformations, where the local influence of a transformation is depended on the distance to the transformations fix point. This leads to local deformations where $T(\mathbf{x}) = \sum_i w_i(\mathbf{x})\ T_i(\mathbf{x})$ with $\sum_i w_i(\mathbf{x}) = 1$ changes smoothly [9]. While the approach with separate layers allows for discontinuities in the motion the second approach does not handle discontinuities but results in an overall smooth transformation.

2.3.2 Image Blending

Image blending techniques combine two or multiple images into a single result. As part of image morphing, image blending interpolates between the appearance of the input images after they have been warped to interpolate the shape or geometric features.

The influence of each image is hereby specified by the distance ratio of the current in-between image α, Equation 2.2. The simplest blending scheme often employed for image morphing is the simple cross-dissolve:

$$B(I_1, I_2, \alpha) = I_1 \ (1 - \alpha) + I_2 \ \alpha \tag{2.10}$$

Surprisingly, this simple approach often yields already very high quality results that don't need to be improved further for image morphing. However, there are two problems where a spatially-varying and non-linear image blending is advantageous. The first type of artifacts during interpolation is due to insufficiently accurate warping of the images to bring their geometric features into correspondence. Despite all efforts, some features may be missed or impossible to be brought into correspondence due to restrictions in the used image deformation model. In this case non-linear blending methods can be employed to correct for some of these errors during blending, e.g. as in our method described in Chapter 6. The second is to cope for regions that are visible in one of the images but not in the second one. When interpolating between images where parts are occluded, then some information is missing in one of the images and thus creates artifacts when blended linearly. This can be addressed by employing a spatially-varying blending scheme as described in Chapter 6 and 7.

3

Related Work

The creation of photo-realistic image sequences is a fundamental goal of computer graphics. The approaches to achieving it can be divided into two categories: To model the world mathematically with sufficient precision such that rendering algorithms can create realistic images from scratch. Or to use photographs and video footage directly such that the inherent photo-realistic quality is preserved while the content is manipulated and combined to achieve the desired results. However, often combinations of both approaches achieve the best results. Especially, the imperfections of reality, the complexity of light transport and complex non-rigid deformations are hard to model sufficiently accurate and computational intensive to reproduce with pure mathematical models. Although we are focusing on image based methods in this thesis, we will also give a short overview on related 3D geometry based methods.

3.1 Natural Phenomena

Natural phenomena are the topic of two chapters in this thesis. Generally, in computer graphics the approaches to model and render such phenomena are divided in simulation and image based methods. The first model the phenomena in 3D using particles, meshes or voxel volumes and apply rendering techniques to create the final results. The second directly reuse recorded images, or create models in image space from recorded footage.

Procedural Modeling A large body of work into realistic modeling and animation of natural phenomena in computer graphics has been concentrating on physical or procedural models in 3D. Techniques following the first approach are based on the use

of particle systems to create explosions for the movie industry such as the work by Reeves [110]. A large numbers of particles are necessary to achieve realistic looking results. However, by careful model parameter adjustment, Lamorelette and Foster [66] obtained very convincing results. The control of these particle systems is dependent on an external vector field, that describes the forces which influence the particles. For example, Beaudoin [8] has proposed a method that simulates the propagation of fire on a 3D mesh.

Simulation Since procedural modeling is based on heuristics, the realism of the final results is strongly dependent on the experience and abilities of the artist. Another direction is thus to directly model the underlying physics, such as the Navier-Stokes equation that describes the transport of fluids and gases, as proposed in the works by Stam and Fiume [133] and Foster and Metaxas [43, 44]. The first fast and stable fluid solver suitable for graphics purposes was introduced by Stam [132]. While this approach gives very realistic results on a large scale, the solver strongly smoothes the solution and important small-scale details get lost. Fedkiw [41] thus introduced vortex confinement that corrects the solution of a simulation time step by adding back lost details. When these simulation methods are combined with the simulation of black body emission and the burning of fuel dissipated by a source, very realistic results are obtained [96]. The drawback of physical simulation is the non-intuitive relation between physical parameters and desired resulting appearance. Often even small changes in the parameters have quite an impact on the results due to the chaotic nature of natural phenomena.

To overcome this limitation Treuille et al. [145] and McNamara et al. [86] solve for external force fields to guide the dynamics of fluids and smoke via key frames. Using their approach physically based animations that are easily controlled are obtained. However, simulation still remains computationally expensive, resulting in an interest in alternative modeling techniques.

Reconstruction Another approach is to record phenomena with multiple cameras and reconstruct a 3D representation. The first work in this direction was proposed by Hasinoff and Kutulakos [53] as a thin 3D sheet with texture. Later, Ihrke and Magnor reconstructed more complex phenomena based on the tomographic reconstruction of

physical properties on voxel grids [4, 58, 59]. While these reconstructions can be used to place real natural phenomena seamlessly into virtual environments, they can not be manipulated or interacted with by virtual objects.

Video Textures The idea of reordering subsequences of videos to produce new output sequences was introduced by Bregler et al. [19]. An input video of a talking person was segmented into phoneme sequences which were then used to produce new sequences of the same person saying different things. For the case of general video a similar approach was proposed by Schoedl et al. [123]. Their idea was to segment video sequences into smaller time periods that can be rearranged to create new and possibly longer sequences. However, this requires that images in the video can be rearranged without noticeable transitions. A later extension of their work then relaxed this assumption by looking at space-time video volumes. Then transitions on a per pixel rather than a per-frame basis are searched for [65]. Finally, they also allowed for incorporating key frame images to script the synthetisation of novel sequences [122].

In the special case of faces, Ezzat et al. [39, 40] have proposed dynamic image space models that are learned from recorded video material. Here the goal is to use visemes (the facial expression related to a phoneme) as key frames and create transitions between these visemes by applying image morphing.

Dynamic Textures Another approach to synthesizing novel dynamic video sequences from recorded videos are dynamic textures by Doretto et al. [35, 130]. They introduced a general framework for statistical learning of temporal regular video sequences. Since the manifestation of the dynamic textures are samples of a spatio-temporal random process, user control of the learned animation is limited to basic properties such as playback speed [36]. Also the realism of the synthesized videos is strongly dependent on how chaotic the depicted object is. This is due to the fact that the method models spatial structures only statistically. Another approach for editing video sequences of natural phenomena is proposed by Bhat et al. [12]. Here the user paints flow lines into the sequence, which are then used to learn textured particles from the input sequence. By rearranging the flow lines they are able to edit such video sequences.

3.2 View and Time Interpolation

View and time interpolation is a method to create views of real world scenes recorded with multiple cameras. Based on the recorded footage and possibly additional information, the goal is to create novel in-between views that have not been recorded directly but are computed by combining the information of the input data. The first type of works achieve the creation of novel images by explicitly reconstructing 3D geometry and textures from the recorded image sequences which are then rendered from novel virtual cameras. To achieve this, the approaches rely on epipolar geometry as defined between two or more perspective cameras. Thus, two assumptions need to be fulfilled by recorded image sequences:

- the camera positions are measured or must be estimated from the images themselves, and
- the cameras are time-synchronized such that corresponding images from different viewpoints show the exact same time instant of the scene

3D Model Fitting For the special domain of architecture Debevec et al. [30] introduced a method based on a coarse geometric model. The coarse geometry is refined with a stereo approach of recorded images and novel views can be rendered using view-dependent texturing. Carranza et al. [22] presented a method for free viewpoint video for human actors by fitting a deformable 3D human body mesh to the images captured with a synchronized multi-camera setup. This is implemented by an optimization approach on the mesh deformation parameters, such that the silhouettes of the resulting pose of the mesh and the result of an image segmentation of the actor where brought into correspondence. For rendering, the mesh is rendered with the pose parameters and projective texturing is applied to recreate the actors appearance.

However, the human mesh model used is generic and could thus only imperfectly approximate the actor of interest. Recently, de Aguiar et al. [29] extended this approach by first 3D scanning the actor in a static pose and then estimating the deformation of this mesh using a similar camera setup. With their approach they achieve high-resolution dynamic meshes of actors with arbitrary clothing.

Image Based Rendering The idea to render complex objects and scenes using only recorded images was introduced independently as light field rendering by Levoy and Hanrahan [70] and lumigraph rendering by Gortler et al. [50]. Both approaches reconstruct the plenoptic function (see also Section 2.1) using a large number of known views. The difference of both approaches is that Levoy and Hanrahan assume a regularly sampled acquisition and create in between views by interpolating a 4D function. Gortler et. al handle unstructured recorded images and resample these in an intermediate data structure, the so called lumigraph. In addition to Levoy and Hanrahan they can also make use of a geometric proxy to improve the interpolation results. Later Buehler et al. [20] introduced the unstructured lumigraph which similarly to the original work by Gortler et al. handles unstructured input views, but does not require the intermediate resampling step. While these works showed only static scenes, somewhat due to the large number of images that must be taken to achieve good quality, special camera hardware has been built to be able to also capture dynamic light fields as described in the works by Wilburn et al. [158, 159]. Another approach based on the reconstruction of the plenoptic function was introduced by McMillan and Bishop [85]. Here the inputs are cylindrical projections taken at different spatial locations and that are interpolated to create novel viewpoints. If in addition per-pixel depth information is available, as for example reconstructed with [142], Schirmmacher et al. [120, 121] showed how to further improve the results while reducing the number of necessary input images.

In contrast to explicitly reconstructing per pixel depths to improve the interpolation results, Seitz and Dyer [125, 126] determine the fundamental matrix to estimate dense disparity and warp-interpolate between two views of a static scene. An extension to dynamic scenes was proposed by Manning et al. [79], segmenting different motion layers by hand and restricting motions to rigid-body translations. A similar approach based on manual segmentation that is also able to address non-rigid deformations was published by Xiao et al. [162]. Later they [163] extended their approach to interpolate between three images, based on the assumption of rigidly moving objects. An automated view interpolation method of static scenes has been proposed by Lhuiller et al. [71].

While these methods interpolate images of both mixed time and view differences, Wang and Yang [153] introduced an extension to light field rendering in general, to deal with non-synchronized cameras. They compute time interpolated images for each

camera to achieve time synchronizity and then use standard view interpolation methods on the time interpolated images to separately interpolate viewing directions.

A related but different application of image based rendering was introduced by Snavely et al. [129]. Based on sparsely reconstructed 3D features from a large amount of images navigation through this image set is implemented by a restricted 3D navigation.

Image Based Reconstruction If additional information about the depth of each pixel is available this information it can also be used to create in between images. The first work to make use of this information was the seminal work of Chen and Williams [25]. They used their approach to achieve interactive viewpoint navigation of complex 3D scenes, that where modeled in 3D but could not be rendered in real-time. Mark et al. [80] also followed this approach but also handled occlusion and discontinuities during interpolation rendering. As for real world scenes, no depth information is available from standard cameras, Zitnick et al. [173] reconstructed this information using a stereo approach. With their reconstruction they were able to create high-quality view interpolations between a set of synchronized video cameras in real-time.

If one is only interested in the reconstruction of a single foreground object that is possibly placed into a virtual environment another approach has been proposed. The visual hull approach of Matusik et al. [84] reconstructs a geometric proxy from a set of sparse calibrated cameras and the segmented silhouette. However, especially incorrect segmentations can lead to an incorrect geometric reconstruction, especially cutting off small scale features, such as fingers or filling in small gaps, e.g. between arms and the body. Goldluecke and Magnor [48, 49] addressed this problem by taking temporal coherence into account and reconstructing a 4D space-time surface. Recently, Starck et al. [134] have also introduced improvements on the geometric reconstruction to achieve high quality and high resolution geometries from sparse synchronized camera setups based on silhouette segmentation.

While these methods improved on the quality of the reconstructed geometry, the quality of the final renderings is also strongly dependent on the appearance or texturing of the model. Especially insufficient camera calibration accuracy and remaining deviations from the true 3D surface lead to artifacts such as ghosting and wrong textures at occlusion boundaries. Eisemann et al. [38] proposed a method based on warping the images of the cameras before projection to correct these problems. Their method

can be performed in real-time on recent graphics hardware and thus instantly improves rendering results.

While some of the previously mentioned methods could create intermediate geometric information, such as the model fitting approaches and the approach by Goldluecke and Magnor, no appearance information can be easily reconstructed. Vedula et al.[150] addressed this problem by reconstructing the 3D scene flow, that is voxels with color and velocities, instead of explicit surfaces. This allowed them to reconstruct also in between images in time and view.

Dense Optical Flow describes a per pixel vector field that describes the motion of each pixel between a pair of images in image space. Thus, this motion is independent of additional information such as camera calibration, synchronizity or scene geometry. The seminal work to compute the dense optical flow with a variational approach was published by Horn and Schunck [55]. There are also some works and web pages that compare and measure the performance of the recent approaches on standard datasets [5, 6, 7]. The basic assumption underlying nearly all dense optical flow estimation methods is the color constancy assumption, stating that all changes between a pair of images can be explained by the optical flow [55]:

$$I_t(x, y) = I_{t+1}(x + \delta x, y + \delta y) \tag{3.1}$$

Highest accuracy has been achieved with variational approaches on multiple scales, such as the work by Papenberg et al. [100]. The main issues that remain is that these optical flow algorithms are inaccurate at preserving motion discontinuities and large occlusions.

Improvement in this direction has been achieved by tackling simultaneously the dual problem of flow estimation and image segmentation. One direction is to pose the problem as the estimation of static image layers and parametric or non-parametric motion estimates for each layer [64, 124, 154]. While these methods can also be used to achieve a higher output resolution as in Schonemann and Cremers [124] or can be used to learn complex models of the depicted objects [64] their application is restricted to short video sequences of dominantly rigid moving objects.

More general solutions are found by not explicitly solving for image layers but by using image segmentation to regularize the flow estimation as in [14, 164, 172]. The

basic idea of these methods is to jointly use the current flow and image values to improve the segmentation while in the next step optimizing the flow for the given segmentation. Zitnick et al. [172] additionally improved this by allowing for overlapping segments and alpha blending. Another improvement was published by Xu et al. [164] where the computing of a confidence map and a parametric motion model is used to correct errors in the estimation.

If more than two images of a sequence are available the estimation can also be improved. Sand and Teller [118] propose to track particles over a video sequence to increase the consistency over time and solve for occlusions.

In practice, optical flow algorithms give very high quality results when the motion is in the order of several pixels. To also handle larger displacements they rely on multi-scale approaches, which on the other hand however requires that objects with large transformations are visible on small scales. Further, general space-time interpolation often implies changes in the images that violate the basic color constancy assumption.

Sparse Optical Flow and Feature Matching Since the dense optical flow can not always be reconstructed reliable due to ambiguities, another approach is to only reconstruct the flow for specific image features. Lucas and Kanade [77] pioneered this approach. Based on a set of point features in the first image, they search for the corresponding features in the second image. Shi and Tomasi [127] published a theoretical derivation of the possibly best features to track between images, based on some restrictions on possible motion.

A similar but more radical approach to match pairs of images uses feature descriptors for matching. Typically these approaches are used to find images that contain a given sample pattern. The matching is then performed on this feature descriptors rather than feature locations. Lowe [76] has introduced the famous SIFT feature descriptor which is invariant under affine transformations. There are also some works on the performance analysis of feature descriptors [88, 135].

If a matching between very diverse images of similar objects is the goal, another approach is to rely on the 2D silhouette or shape of the object for matching. Mori et al. [90] introduced the shape context descriptor for the matching of points on silhouettes. By building relative histograms of the spatial distributions of regular point

samples along the silhouettes, a descriptor is computed which can then be used for estimating similarity during matching. While this approach assumes rigid transformations and only small variations, Leordean et al. [68] extended the matching also for larger deformations. By preserving local distances to the direct neighbors during matching they achieved a good and controllable trade-off between matching performance and shape preservation. Finally, there are related probabilistic approaches to correspondence finding in the 3D reconstruction literature solving the structure from motion and correspondence problem such as [31, 34, 146].

3.3 General Image Interpolation

Image Warping Image warping is the deformation of images from user-defined correspondences. Well-known is the line-based warping method proposed by Beier and Neely [9] from its use in Michael Jackson's music video "Black & White". Lerios et al. [69] extended the approach to 3D voxels and addressed ghosting artifacts by correcting the warp field. Many other warping techniques have been proposed and are summarized in Wolberg's seminal work [160], including the popular thin-plate spline interpolation. A computationally more complex method based on line features that enforces local constraints related to a group of geometric transformations was recently proposed by Schaefer et al. [119]. This approach has been also implemented on the GPU for improved performance and extended to curve features by Weng et al. [155].

The so far discussed methods have in common that they are solely dependent upon a sparse set of user given correspondences and additional control parameters, e.g. the influence and type of the local influence of the correspondences. Thus they are independent of the actual image content and do not make use of this information to automatically improve the interpolation result. Glasbey et al. [46] give a review of parametric image warping methods that extend image morphing into this direction and achieve semi-automatism. They provide a classification of different approaches and discuss a Markov-Chain Monte-Carlo (MCMC) approach for warp parameter estimation with user-provided correspondences. Another probabilistic approach to image warping for elastic image registration is presented by them in [47].

Some fully automatic morphing approaches for special cases exist. Morphing faces has received a lot of interest. Existing automatic approaches can be divided into two

categories. The first makes use of a generic parametric 3D model, which is adjusted to fit the input images [16], [104]. Other approaches propose image based models of faces such as [13], [169] and [72].

Another approach to image interpolation based on image warping and blending especially suitable for sequences of natural phenomena is presented by Chartrand et al. [24]. The idea is to view the warping not only as a deformation of the images but also as a redistribution of the overall intensities by solving the Monge-Kantorovich problem [89]. However, a drawback of the method is that the images have to be normalized before the warping can be computed. This sometimes results in unwanted image alterations. A recent improvement in computing the optimal image warping in the Monge-Kantorovich sense is published by Haker et al. [51]. The authors suggest a very fast local approach to solve the problem.

3.4 Video Synchronization

Over the last years, the problem of finding the temporal offset between multiple recorded video sequences recorded with unsynchronized video cameras has been addressed by many researchers. The proposed approaches can be roughly classified in two categories depending on the goal of achieving frame or sub-frame accuracy.

Like the methods by Yan and Pollefeys [165] and Spencer and Shah [131], the first class of methods find the integer frame offsets between unsynchronized cameras. In [165], points with spatio-temporal variations are detected in the video sequences and are described as a temporal distributions. If the dynamics of the scene are similar in the recorded cameras, their distributions are similar and the temporal difference between the sequences can be calculated through an alignment. In [131], the movement of the objects is analyzed and compared, which allows to synchronize camera streams of different scenes as long as they have the same dynamic.

By contrast, [157] and [23] achieve sub-frame accuracy by calculating the fundamental matrix of the cameras and by evaluating it afterwards on the basis of correspondences between trajectories of moving objects. While Whitehead et al. [157] propose to use three cameras and to calculate the trifocal tensor, Caspi et al. [23] just need two. Both approaches are however limited to stationary or jointly moving cameras. In Dai et al. [28] an iterative algorithm is presented using 3D phase correlation based on

a projective geometry constraint. For this purpose, the simplified assumption is made that the centers of the cameras are comparatively close to each other and, therefore, parallax can be neglected.

3.5 Perceptual Adaptive Graphics

Many graphics areas can benefit from related literature of perception both in improving the overall quality as well as in focusing on important features. A very good survey of interesting perceptual models which have been applied successfully in computer graphics has been published by O'Sullivan et al. [98].

Recently Vangorp et al. researched the influence of material of objects to shape perception [148, 149] showing interesting dependencies between differentiability of material properties and surfaces in extensive user studies. Ramanarayanan et al. conduced user studies on visual equivalence of rendered images [109] and the equivalence of the rendering of groups of objects depending on shape and texture [108]. Both approaches open new insights on where computational power can be saved, in form of a simplified material and/or shape properties or reduced object numbers, if the human observer is understood as the consumer of the output of the rendering pipeline.

Another topic that is based on perceptual models and that has received very much attention over the last years is the display of high-dynamic range image content (HDR) on low-dynamic range display devices [113]. For example the well-known methods presented by Pattanaik et al. [101] and Reinhard and Devlin [112] are based on such models.

4

A Dynamic Image Space Model for Flames

4.1 Introduction

In this chapter we deal with the rendering and animation of flames. The dynamics and motions of flames are governed by hydro-dynamical forces. The interaction between laminar and turbulent flow within flames can cause not quite periodic, yet at the same time also not-yet truly chaotic flame motion. The appearance on the other hand is defined by the black body emission of soot particles.

The rendering and animation of realistic flames can be implemented based on physical models. By careful model parameter adjustment, the obtained results are very convincing [66, 96]. However, fine-tuning the model parameters towards some desired flame output is tedious as the parameters' physical interpretation does not directly relate to their impact on flame animation and appearance. Another drawback of physically based approaches is the computational power needed to solve the underlying differential equations which necessitates trading off animation realism for real-time performance.

A different approach to rendering flames are image based methods. Such approaches have been successfully applied to prolong and loop recordings of recorded flame sequences [123, 130]. However, these methods allow only restricted control of the output or require a large amount of images to achieve suitable flexibility.

In this chapter we follow the second approach and deal with the rendering and

animation of flames in image space. Our approach is based on machine learning to capture the dynamics and appearance of recorded flame sequences. The learned flame characteristics are sufficient to render plausible new flame sequences in real-time. With our method we are able to synthesize arbitrary long and unique flame sequences that are yet similar to the (much shorter) input sequence. The underlying model allows also for extensive control and interactive manipulation of the flame in real-time.

4.2 Flame Appearance

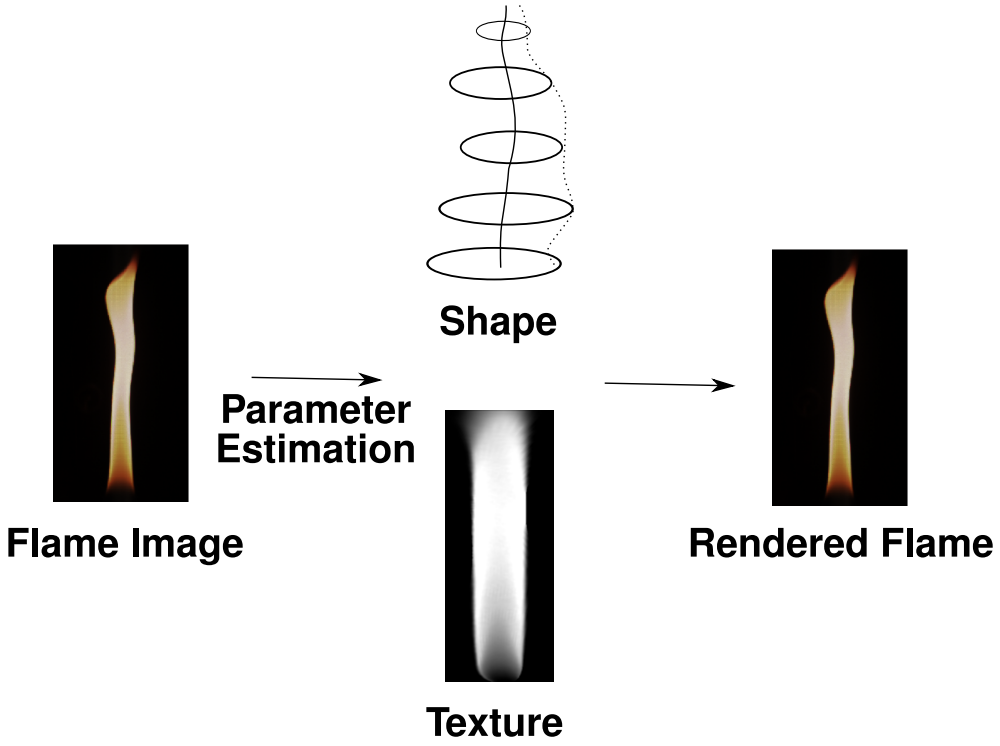


Figure 4.1: Our proposed flame model. The model consists of shape and texture parameters which are both estimated automatically from recorded flame images. The estimated parameters are sufficient to render novel flames.

An image space model that can be used to estimate and render images from a dynamic sequence has to model both appearance as well as dynamic properties. While the underlying physics of flames and fire in general are quite complex and produce very different shapes, the shape of a single flame can be seen as an approximate geo-

metric primitive, e.g. as deformed cylinder with varying thickness [8]. While such an approximation captures sufficiently well the coarse shape of a flame, it is however not sufficiently accurate to render plausible new flame images, since the distribution of the soot particles within the flame is neglected. Further, typical flames have no well defined surfaces but rather fuzzy boundaries. Thus in addition to the coarse geometrical properties, we analyze texture variations with a principal components analysis (PCA), to capture the most significant variations in an optimal image basis. The analysis of the rather complex flame dynamics is then simply the analysis of trajectories in the space defined by our flame model over time.

4.2.1 Flame Shape and Texture

We model flame appearance in image space by two different sets of parameters. First we approximate the coarse shape by a deformed cylindrical primitive. The approximating cylinder can be described by a central axis and widths along this axis as depicted in Figure 4.1. In addition to this coarse shape model, the remaining texture variations on the per pixel level must be modeled. The naive approach to use the images directly would yield a high quality representation. However, one can do better in terms of necessary space consumption and exploit the fact that flames are rather structured objects. Principal components analysis (PCA) has proven a powerful tool in dimensionality reduction of images depicting the same or a similar object, as has been successfully shown for face images by Turk et al. [147]. However, PCA can only be successfully applied if the images are properly geometrically registered, so that the variation of the intensity of each pixel derives from a change of texture rather than a change of shape. With our coarse parameters derived from the approximate deformed cylinder model we can factor out the geometric variation. Thus by using the shape model to project the image of a given flame onto a *shape-normalized* flame the remaining variation is just this variation in texture.

4.2.2 Estimating Shape Parameters

As discussed in the previous section, we model shape with a deformed cylinder. To robustly estimate the parameters of the approximating cylinder from the image of a

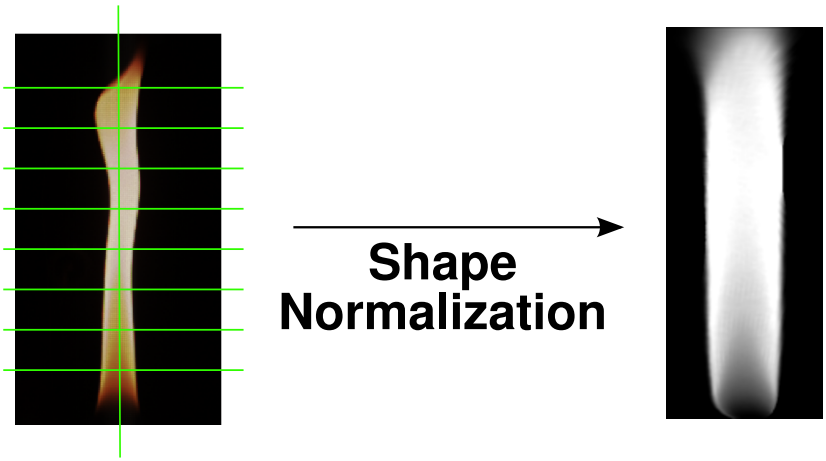


Figure 4.2: Shape normalization of the recorded flame images is achieved by estimating the shape parameters with image moments. After the shape normalization process the remaining variance among a set of flame images is only due to variations in texture.

flame, we make use of geometric image moments M_{pq} with

$$M_{pq} = \int \int x^p y^q I(x, y) dx dy, \tag{4.1}$$

where I is the gray valued image of the flame and x and y are the coordinates [93]. The order of the moment is defined as $p + q,\ p, q \in \mathbb{N}$. We use the computed moments up to order three to shape normalize the recorded flame image. Specifically, after the first normalization step the following properties hold

- all flame images exhibit the same average intensity.
- the axes of the cylinder approximating the flame are aligned with the image axes.

Then we can decompose the flame images along the y-axis into horizontal slices and measure the eccentricities and widths according to the deformed cylinder model with one-dimensional image moments of each slice, cf. Figure 4.2. Together with the flame's height we obtain N_p+N_w+1 parameters to describe the shape of each flame. Finally, the flame image is deformed by a thin-plate spline warping of the flames shape coordinates on to the normal flame shape as depicted in Figure 4.2. The resulting flame images are stripped of all shape information while the texture information is still preserved.

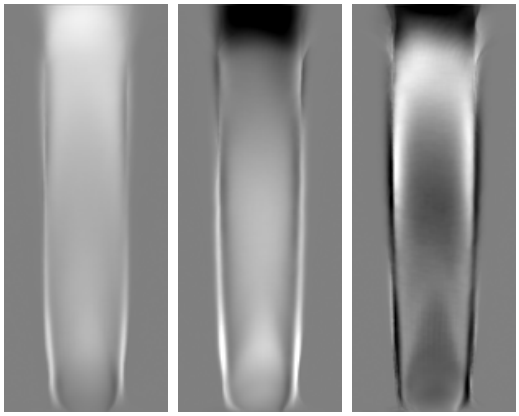


Figure 4.3: The three eigenflames associated with the largest eigenvalues of the PCA analysis of the recorded sequence. These eigenflames represent appearance variations of the flame, e.g. overall brightness (left), and variations in the upper (middle) and lower (right) part of the flame.

4.2.3 Estimating Texture Parameters

After the flame shape parameters have been estimated and the flame images have been shape normalized, the variation in texture must be addressed to achieve plausible results when rendering novel flames. To do this we follow a recent approach in texture analysis with PCA [147] on all shape normalized images of a recorded flame sequence. Specifically, we compute a set of *eigenflames* that forms a basis for our texture flame space, Figure 4.3 as follows: Given a set of N shape-normalized flame images I_i we first compute the mean flame $I_{mean} = \frac{1}{N}\sum I_i$ and the covariance matrix of $\Sigma = \frac{1}{N} X\ X^T$ where column i of X is composed of the elements of $I_i - I_{mean}$. By calculating the eigenflames and eigenvalues of Σ we find a new image basis for the normalized flames. In this new basis we then drop all but the dimensions associated with the N_e largest eigenvalues. This results in a significant reduction of data necessary to represent the normalized flames, while still covering the essential texture variations. In this image basis, each flame texture is then approximated by the weighted linear combination of the eigenflames according to its N_e texture parameters.

Note, that the proposed flame model is useable for analysis and synthesis. This means we can automatically estimated its parameters from a recorded flame image *and* recreate an approximated but very similar image of the flame, cf. Figure 4.1.

Further, novel flame images that have not been recorded but are still plausible are just coordinates in the space spanned by this model.

4.3 Flame Dynamics

So far we have focused on how to describe the appearance of a flame at a single point in time as a point in the space spanned by our flame model, cf. Figure 4.1. However, we are also interested in the dynamics of a burning flame over time. Expanding the proposed flame appearance space into the time dimension, the dynamics of a burning flame then forms a trajectory in flame space over time. From the quasi-periodic dynamics of burning flames, we conclude that the dynamics can be approximated with an stochastic auto regressive processes (ARP) [75].

In the following, we refer to the model parameters derived from the input video sequence as $\mathbf{x}_t$. Thus, $\mathbf{x}_t$ is a vector of length $N_p + N_w + N_e + 1$. Note, that at time instant t, the flame is completely represented by $\mathbf{x}_t$. We assume the temporal Markov property which renders the model independent of any previous state $\mathbf{x}_{t-k}$ for $k > K$. This leads to the definition of the ARP as follows:

$$\mathbf{x}_t = \sum_{k=1}^{K} A_k \mathbf{x}_{t-k} + \mathbf{d} + B\mathbf{w}_t \tag{4.2}$$

where A_k and $\mathbf{d}$ are deterministic parameters of the process and matrix B models the weighting of the stochastic vector $\mathbf{w}$ where each component w_i is an independent random variable with normal Gaussian distribution. An ARP is now sufficiently described by the set of parameters $\lambda = \{A, \mathbf{d}, B\}$ with $A = (A_1, A_2, ..., A_K)$. In what follows, we are searching for a parameter set λ^* that best represents the recorded flame dynamics.

In addition, we assume that the state of the model $\mathbf{x}_t$ is not observed directly but is the result of a noisy observation. This assumption reduces the influence of errors in the estimation of the parameters in the previous step. Since the state of the model is hidden, using a Maximum-Likelihood (ML) approach to estimate the parameters is not directly possible [45]. Instead, we use an iterative Expectation-Maximization (EM) algorithm for learning the stochastic dynamic process [32].

EM with Condensation To learn the ARP, we resort to the condensation algorithm introduced by Isard and Blake [61]. The condensation algorithm is, in essence, a particle filter approach which approximates a multivariate distribution. In the context of learning an ARP, the condensation algorithm was first applied in the expectation step of EM algorithms [15, 97]. We give a short outline of the learning process here.

The EM algorithm is divided into two steps. The first step consists of estimating the expected values of the process under the assumption of noisy observation given an ARP λ_{i-1}. In the second step, an updated ARP λ_i that maximizes the likelihood of the expected values is computed [75]. Both steps are repeated until convergence.

As mentioned above, the condensation algorithm is used to estimate the expected values in the expectation step. The number of particles of the filter used in the computation is set to N_c. Because the approximation quality of the true multivariate distribution increases with N_c, one has to find a reasonable trade-off between computational complexity and goodness-of-fit. In our experiments we found that a value of N_c in the range of 50 to 300 is a suitable choice.

Initialization Since the EM algorithm converges to a local optimum, the learning results depend on a good initial ARP λ_0. For the computation of λ_0 we simplify the problem in two ways. First, we restrict the degrees of freedom of the initial ARP by assuming mutual independence of the flame model parameters which means all A_k^0 are diagonal matrices. Second, with the assumption that $\mathbf{d}^0 = 0$, we can estimate A_k^0 directly from the input data. B_0 is initialized as a diagonal matrix with standard deviations of the observed parameters on the diagonal.

4.4 Rendering

Any point in our flame space describes a flame on the per pixel level. Thus, we can render novel flame images related to any point in flame space, even the ones that have not been recorded. Figure 4.6 shows some of the novel images of flames that have not been recorded but are still plausible. Having learned the ARP that represents the characteristic dynamics of our flame, each manifestation of the stochastic process forms a new trajectory in flame space. This is done by repeatedly evaluating (4.2) to compute

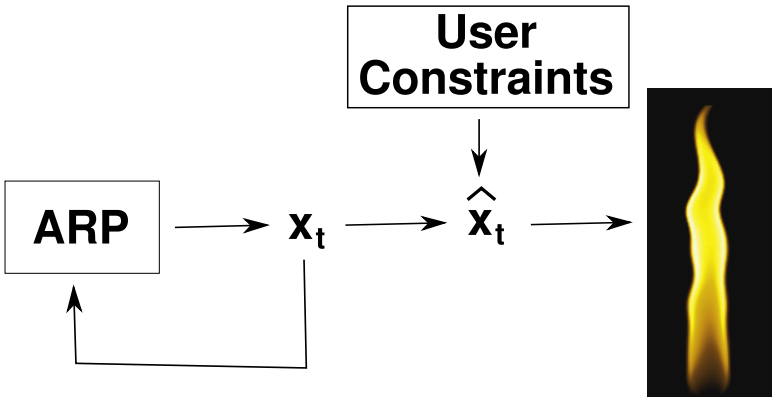


Figure 4.4: To generate a sequence of new flames we sample the learned ARP. Novel flame images of the samples $\mathbf{x}_t$ are rendered using the image space flame model.

the next $\mathbf{x}_{t+1}$. Figure 4.4 gives an overview of the rendering and sampling process to create arbitrary long novel flame video sequences.

While the flame model so far specifies gray scale flame images indicating emissivity, we can also compute a color interpretation by approximating the underlying physics. Due to the nature of flames we can introduce a transfer function to map flame intensity values to color values associated with temperature from blue to yellow.

4.5 Results

The presented results are based on a sequence of flame images recorded with a 1 Megapixel camera at 40 fps, Figure 4.5. Recording the video sequence in front of a dark background avoids segmentation of the recorded flames. From the recorded video sequence, images of flames consisting of two or more disjoint plumes are discarded, because our flame model, Section 4.2, cannot represent such (rather rare) flame appearances. From the recorded video sequence, a sub-sequence of 100 flame images was selected that shows a single flame without topology changes, i.e., without plumes. Figure 4.5 depicts three images from the input sequence. The used flame model consists of $N_p = N_w = 10$ and $N_e = 5$ parameters yielding a total of 26 parameters. For the initial ARP we use our proposed heuristic and set the Markov time horizon to $K = 2$. $N_c = 200$ particles for the condensation algorithm are used. The learning of the ARP is terminated when the likelihood does not increase during three consecutive iterations. Our Matlab im-

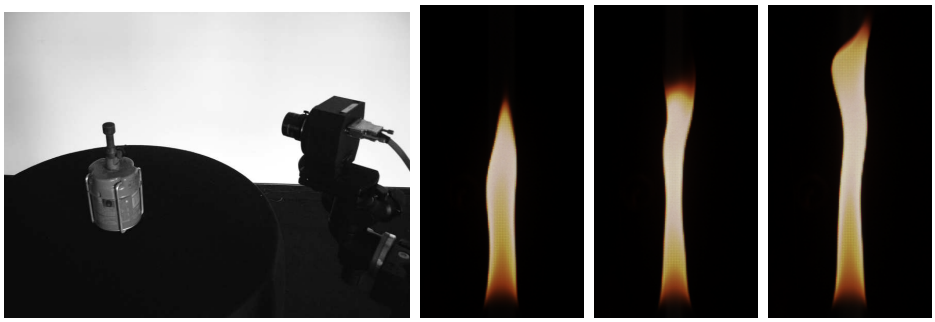


Figure 4.5: The acquisition setup used for our experiment and three images from a recorded flame sequence. The flame was produced by a Bunsen burner and was recorded with a 1 Megapixel camera at 40 Hz sampling rate.

plementation takes about 30 minutes on a 3 GHz Intel Pentium 4 to find the ARP for the recorded sequence.

4.5.1 Control and Interaction

Figure 4.6: Novel flames created from the learned model.

Arbitrary new, unique flame sequences can be synthesized from the learned model in real time. Figure 4.6 depicts novel flames rendered from samples of the learned ARP. Note, however, that the true quality of the proposed algorithm can be assessed only from the animated flame.

While a synthetic flame with the same flickering and movement characteristics as the original flame can be obtained directly without any additional scaling or transformation of the parameters, our model additionally gives control over the flame appearance. For example, the shape parameters specifying the center of the cylinder approximation of the flame can be adjusted over time, e.g. by adding an offset to the path parameters, resulting in a bent flame. This can be useful to introduce the effects of external forces such as wind. The same direct manipulations can be applied to the flame height, width, and scale. This gives an artist easy control over the rendered flame results.

4.6 Summary

In this chapter we have introduced a novel approach for the animation and modeling of flames. We propose a image space flame model which can be robustly estimated from recorded video sequences. By learning the temporal characteristics of the flame shape and texture, we are able to synthesize arbitrarily long, unique sequences[1] in real-time.

Through the combination of a statistical approach for the temporal behavior and the image based approach for appearance of the flame, interesting manipulations of the flame are possible. For example, flames can be bent or deformed interactively while the overall burning characteristic is still maintained. Physical correct interaction of the flame with the environment however is up to the animator.

[1]The length without repetition is in fact dependent on the performance of the random number generator used. However, for any practical application no repetition will be noticeable by a human observer.

5

Key-frame Animation of Natural Phenomena from Video Sequences

5.1 Introduction

In the last chapter we introduced a dynamic image space model for burning flames. In this chapter we will propose a dynamic image space model for natural phenomena with quasi-periodic character by combining two interesting and powerful recent approaches to tackle this problem. In a first step, we address the problem of selecting parts of the videos that show a periodic character by visualization and user interaction. The proposed view reveals such periodic parts as loops in the video trajectory, shown in Figure 5.1. In a second step, these subsequences are then reordered and repeated automatically or by an artist to synthesize new sequences. To smooth the transitions between reordered subsequences, we make use of an image interpolation method especially suited for images depicting natural phenomena

5.2 Video Analysis

In the first step a recorded image sequence is analyzed to reveal its periodicity. For example, a burning flame sequence exposes such character as the flame appearance typically varies in a periodic manner. To achieve this, we follow a recent approach by Pless [105] for visualizing image sequences as video trajectories. Such a visualization in

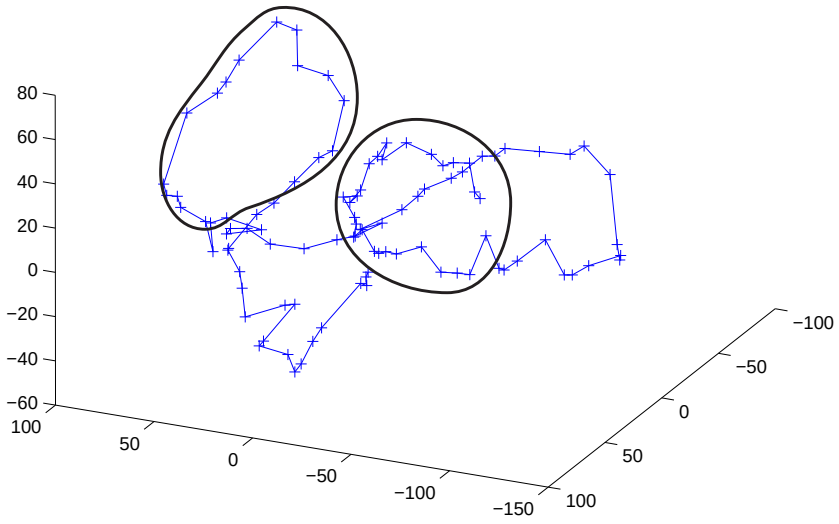


Figure 5.1: Low-dimensional representation of a burning fire sequence. The Isomap algorithm represents all images as points in a k-dimensional space (here k=3) so that the distances between the images are preserved up to a minimal residual. Connecting the images according to their temporal order results in a video trajectory. The loops that can be seen in this view identify periodic subsequences.

3 dimensions of a sequence depicting a burning fire is shown in Figure 5.1. As can be seen in this example, the periodic character of the input sequence is intuitively revealed by the loops of the trajectory.

5.2.1 Video Trajectories

The visualization of the video sequences of natural phenomena is based on an representation in a 3D cartesian coordinate system which preserves an image metric $D(I_i, I_j)$ between all pairs of images in the sequence. This means, we search for a 3D coordinate to each frame of the video sequence such that the L_2 Norm of the associated coordinates approximates the given metric between the pairs of images. The Isomap algorithm presented by Tenenbaum et al. [144] finds such an mapping by nonlinear dimensionality reduction. The algorithm takes as input the distances between points

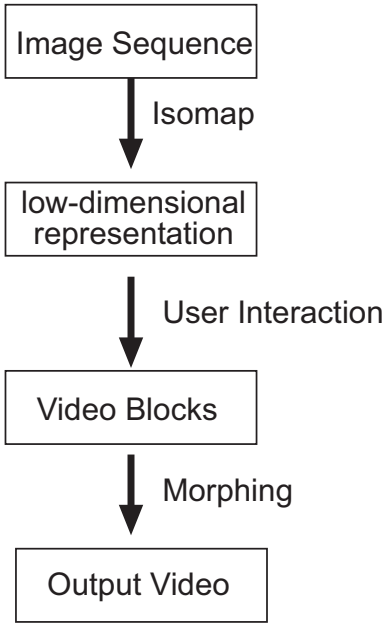


Figure 5.2: Overview of our method. From an input video sequence, first a low dimensional representation is obtained based on the Isomap algorithm. Using this representation of the video the user can identify quasi-periodic snippets. These subsequences are then reordered either automatically or user-driven to create new sequences. Transitions between the blocks are smoothly interpolated using Monge-Kantorovich based image morphing.

in a high-dimensional observation space, and returns as output their coordinates in a low-dimensional embedding that best preserves their intrinsic geodesic distances. Once such an embedding is computed, that is we have obtained cartesian coordinates for each video frame, we can plot the video as a 3D trajectory as shown in Figure 5.1. For the natural phenomena sequences we analyzed, this representation reveals the quasi-periodic subsequences as loops in the trajectory if the image metric for the embedding is chosen appropriately.

The simplest sensible distance metric between images are the sum of squared distances. The advantage of this metric is that it is both simple to implement and can be computed very fast which is crucial as the number of pair distances is growing exponentially with the number of frames. Interestingly, more adequate measures like the Earth Movers distance [116], result only in marginal improvements that are however not crucial if only the visualization of the video is considered [105].

5.2.2 Video Blocks

In the low-dimensional representation described in the previous section, the user can easily identify periodic parts of the input video. These parts form loops in the trajectory, which are not loops in the strong sense but points on the trajectory that are neighbors in this space (see Figure 5.1). Selecting the neighboring points defines a start and end image of a quasi-periodic subsequence of the video. We denote these subsequences as video blocks to express their role as building blocks in the synthesis step. A video block B is thus a temporally ordered set of images $B = \{I_i : s \leq i \leq e\}$ defined by a start and end image of the subsequence.

5.3 Video Synthesis

After the input sequence has been segmented into video blocks, these are used to synthesize new image sequences. The rearranging of blocks however often results in noticeable temporal discontinuities at transitions between the end image of one block and the start image of the next block. To address these artifacts and create plausible transitions novel in-between images must be computed. A straightforward solution is to use the nearest neighbor image for a given point in the embedded space, similar to the video texture approach by Schoedl et al. [123]. However, new connections between video blocks may have no in-between images in the original sequence. Thus, we apply image interpolation to create new in-between images. Specifically, we use a Monge-Kantorovich based image morphing approach that is especially suited to morph images depicting natural phenomena. The synthesis step is divided into two parts. First we sequence the previously found video blocks. In the second step, new in-between images are synthesized to create smooth transitions.

5.3.1 Sequencing

To create new image sequences, the video blocks identified in the video analysis step are rearranged. We propose several ways to create new sequences from the video blocks. An automatic approach is to specify the probability for each video block. This could be e.g. an uniform distribution if no preferences are given. A more controlled output is achieved by specifying transition probabilities $P(i, j)$ between blocks. This probability matrix P then defines a first-order Markov process which can be sampled to achieve an

automatic sequencing. For full control over the output the blocks can also be sequenced manually by the user.

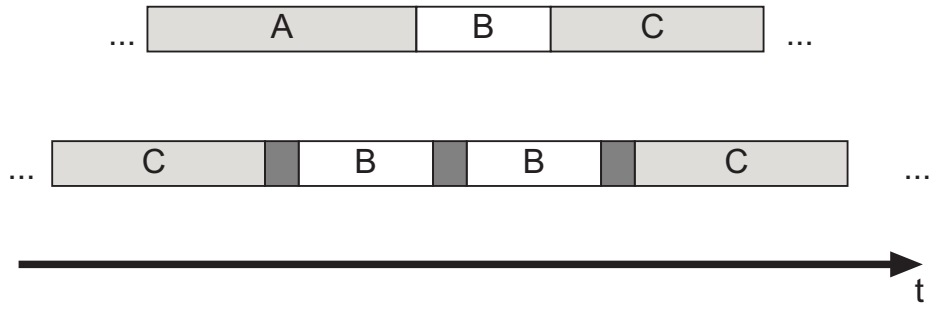


Figure 5.3: Re-sequencing the video blocks labeled A,B and C. To smooth the transition between the blocks new in-between images between the closest images are computed (dark grey blocks). The duration of the blocks can also be adjusted to create slow motion versions of the subsequences.

5.3.2 In-between Images

It is necessary to compute novel in-between images to achieve plausible transitions between reordered video blocks. To do this, we restrict ourselves to automatic image interpolation techniques that need no user interaction to identify matching features. One possible solution are optical flow based interpolation methods as the one by Papenberg et al. [100]. However, for image sequences depicting natural phenomena like fire, the assumption of constant pixel intensity over time is strongly violated leading to unsatisfactory results. Instead, we resolve to image warping methods based on the Monge-Kantorovich problem. Originally stated by Monge [89] for solving the problem of moving piles of dirt in an optimal sense, recent work by Haker et al. [51] and Chartrand et al. [24] has translated this to the problem of image warping. This approach assumes that the *accumulated* image intensity stays constant, unlike optical flow computation which assumes that each pixel intensity stays constant. The Monge-Kantorovich solution is then a transport mapping that redistributes the "mass", which is equivalent to image intensity in the image warping case, in a minimal distance sense. Thus, small bright regions can also be mapped to larger dimmer regions, better reflecting the physical properties of diffusion processes.

Figure 5.4: Monge-Kantorovich image morphing between two images of a flame sequence. The left image is the source and the right the target image. In between is the interpolated image halfway from source to target.

Figure 5.4 shows the morphing results between two images of a fire sequence. The the Monge-Kantorovich warps between the source and the target images are linearly cross-dissolved for computing the shown in-between image.

5.4 Results

We have applied our approach to several real-world video sequences depicting natural phenomena. The example sequences consist of 70 to 200 images in the input sequence. We visualized the sequences with a three dimensional representation based on the L_2 distance between the image pairs to identify periodic subsequences. Sequences showing a single phenomena, like the fire sequence, are best suited for our approach since the periodic character is very prominent. Once the warpings between the selected video blocks are pre-computed the creation of new sequences is possible in real-time.

5.5 Summary

In this chapter we proposed a method for analyzing and synthesizing video sequences, specifically suited for image sequences of natural phenomena with quasi-periodic char-

Figure 5.5: Examples of our test sequences. Sequences showing a single phenomena are best suited for our approach since the periodic character is very prominent.

acter. We introduced a visualization that allows users to easily select parts of the video that show such periodic character. These subsequences can then be reorderd and repeated automatically or by an artist to create the desired output. We achieve smooth transitions between the reordered subsequences by computing in-between images using image interpolation. The interpolation is based on the Monge-Kantorovich based image warping method. In contrast to the standard optical flow this approach is especially suited for diffusion processes since it can map small bright areas to larger dimmer areas in contrast. Our approach is useful for video key-frame animation and automatic looping of image sequences showing natural phenomena.

5. KEY-FRAME ANIMATION OF NATURAL PHENOMENA FROM VIDEO SEQUENCES

6

Image Morphing for Space-Time Interpolation

6.1 Introduction

In this chapter, we look at the problem of space-time interpolation based on image morphing between multi-view video. So far, image based approaches to interpolation of time-varying scenes require that the acquisition cameras must all be synchronized to be able to relate images via epipolar geometry. This need for calibrated, synchronized acquisition is highly inconvenient as it implies time-consuming recording preparations as well as special acquisition hardware. Instead, a general image interpolation approach is able to provide plausible interpolation results across space and time from nothing more than a collection of unsynchronized, uncalibrated images. In contrast to previous work done in image based rendering, we do not enforce physical correctness but optimize for plausibility. Based on the feature based image morphing method by Beier and Neely [9], we introduce several optimizations to improve its suitability to this task: First, we show how the optimization of feature weights and optical flow correction can be used to improve the warping and reduce the necessary amount of user specified features. Then, we discuss how the visibility of remaining artifacts can be reduced by a non-linear blending approach taking human perception into account. Finally, we introduce a shape-preserving interpolation scheme of the line features to achieve plausible interpolation results for space-time interpolation of uncalibrated and unsynchronized multi-view recordings.

6.2 Improving Feature-Based Warping

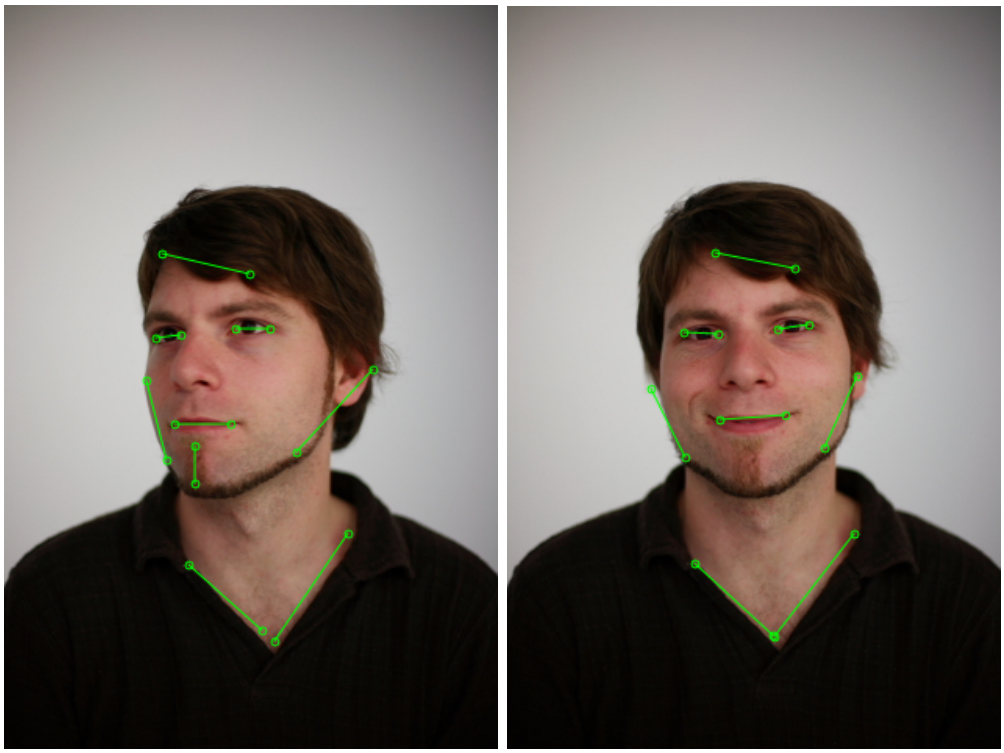


Figure 6.1: Corresponding line features between source and target image can be placed manually and propagated automatically, allowing large numbers of images to be matched in a short time.

The warping method introduced by Beier and Neely [9] relies on manually specified line feature correspondences, cf. Figure 6.1. Each of the line feature pairs defines a similarity transformation between the images (cf. Section 2.3.1). The final image deformation is then a weighted combination of all similarity transformations defined by the line feature pairs. The local varying weights are hereby defined by a weighting function for each line feature, that is dependent of the distance to the line feature. In the rest of the section, we will introduce improvements to the original method. However, before we address these, we first recap the definition of the warping in more detail and introduce a mathematical formulation in homogeneous image coordinates.

Given the two images I_1, I_2, and a set of K pairs of corresponding lines $\Lambda = (l_k, l'_k)$, each mapping defines a similarity transformation T_k represented by a 3×3 matrix in

homogeneous image coordinates between the images. The function w_k, dependent on the Euclidean distance $d(\mathbf{x}; l_k)$ between a given point $\mathbf{x}$ and the line feature l_k, defines the weights used in the interpolation of each linear transformation. Then, the point $\mathbf{x}_1$ in I_1 corresponds to $W(\mathbf{x}_1) = \mathbf{x}_2$ in I_2 as follows:

$$W(\mathbf{x}_1) = \frac{1}{\sum_k w_k} \sum_{k=1}^{K} w_k(\mathbf{x}_1) \, (T_k \mathbf{x}_1) \tag{6.1}$$

with the feature weighting function

$$w_k(\mathbf{x}_1) = \left(\frac{|l_k|^{p_k}}{a_k + d(\mathbf{x}_1; l_k)}\right)^{b_k} \tag{6.2}$$

and weighting parameters Ψ with $\Psi : p_k, b_k \geq 0$ and $a_k > 0$. In the original work, the weighting parameters Ψ are also specified by the user to define the shape of the weighting function, and all line features have the same weighting parameters. While it is stated that the influence of the weighting parameters is neglectable [9, 143] as it can be compensated by specifying more features, this is not beneficial if the interpolation of many images is the goal. We show in the next sections, that by finding per feature optimal weighting parameters and additional per-pixel corrections less features are needed while the quality of the warping result is improved.

6.2.1 Per-Feature Optimal Weighting Parameters

Our first step in improving the method of Beier and Neely is to find the optimal per-feature weighting parameters $\hat{\Psi}$ automatically for a given set of line feature correspondences. The line features might be placed manually or semi-automatically by using automatic propagation methods such as proposed by Szewczyk et al. [143]. Since we want to achieve a plausible transition between images of the same scene seen from different view and/or time points, we define optimality as distances between the halfway warped images $W(I_1, 0.5)$ and $W(I_2, 0.5)$. However, we relax the assumption that the warping brings all pixels of the images in correspondence to each other. Such deviations are frequently observed, typically due to partial occlusions or changes due to non-lambertian surfaces. Instead, we reduce the influence of the differences in image regions that are not close to any feature as follows:

$$\sum_{x \in I_1} \{||(W(I_1, \mathbf{x})) - W(I_2, \mathbf{x}')||^2 \sum_k w_k(\mathbf{x})\}. \tag{6.3}$$

We solve this non-linear optimization problem efficiently with a gradient decent approach [106] on a multi-scale Gaussian image pyramid. In contrast to the original method, this yields already an improvement in the quality of the warping as depicted in Figure 6.2. Additionally, the need for user interaction is reduced since less features are necessary and the weighting parameters are found automatically.

6.2.2 Per-Pixel Warp Field Correction

The per-feature optimized warping still produces small but noticeable artifacts, especially at silhouettes, Figure 6.3. This is due to the definition of the warp field by line features. Only the parts of the image that are approximated with sufficient line features are correctly transformed. This is especially critical at curved contours. The straight forward solution is again to use more line features to better approximate the contours at the pace of more time-consuming user interaction. Instead, we propose to correct the warping function W with an additive per-pixel warping correction term $W_{Correction}$

$$\hat{W} = W + W_{Correction}. \tag{6.4}$$

Again, we want to find $W_{Correction}$ so that differences between the warped images, $\hat{W}(I_1) - I_2$ are minimized. In fact, there is already a large body of work in computer vision, dense optical flow estimation, which solves exactly this problem. Thus, we can resolve to an algorithm that automatically computes the optical flow estimation from the warped images.

In general, optical flow estimation performs very well for images that are suitably similar to each other. However, occlusions and large changes often also cause problems in regions that could be matched correctly, as the general assumption of color constancy is strongly violated. Starting from a warped pair of images however, the mentioned problems of optical flow estimation are effectively circumvented because the warped images are already coarsely aligned. Thus, we compute $W_{Correction}$ as a linear combination of automatically computed dense optical flows OF between the warped images:

$$\begin{aligned} W_{Correction}(x,t) = t\, OF(W(I_1,t), W((I_2, 1-t)) + \\ (1-t)\, OF(W(I_2, 1-t), W(I_1,t)) \end{aligned} \tag{6.5}$$

With this per-pixel correction of the warping we can significantly reduce artifacts due to small image mismatches and missed details, Figure 6.3.

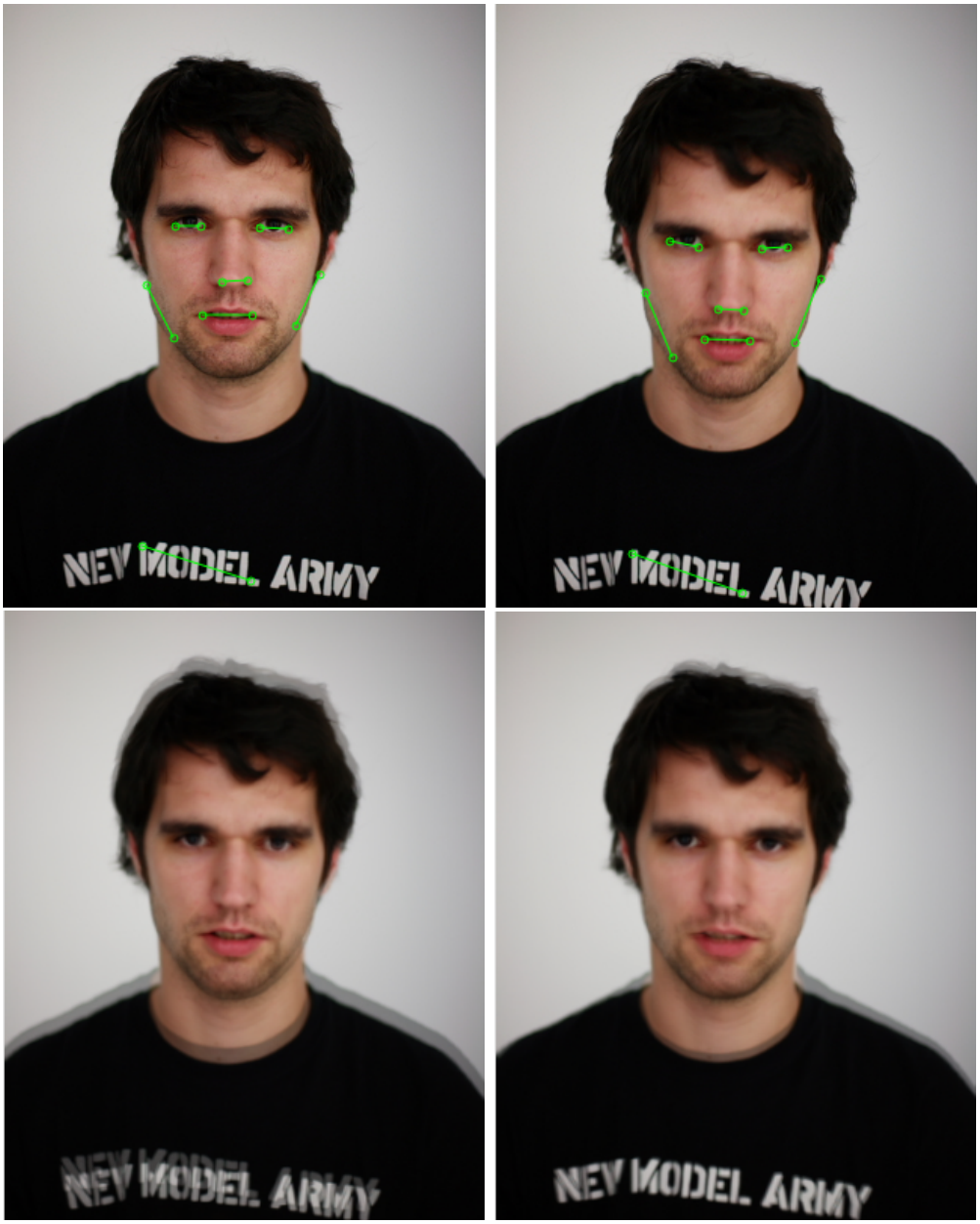


Figure 6.2: Comparison between morphing results of the original method by Beier and Neely (bottom left) and our proposed optimization and generalization approach (bottom right) at $t = 0.5$. The input images with the given line feature set Λ are depicted in the first row. We achieve a signification reduction of the ghosting errors especially noticeable around the text logo.

Figure 6.3: Comparison of the morphing results of the same in-between image: without (left) and with (right) per-pixel warping field correction. Artifacts due to contour mismatches are significantly reduced.

6.3 Perception-motivated Non-linear Blending

Our goal is to achieve plausible transitions between images showing the scene at different time and/or viewpoints with an image morphing technique. Although the optimizations in the warping already improved the interpolation results, there are still some artifacts remaining which cannot be addressed by means of warping alone. Specifically, the most prominent remaining source of visible artifacts is due to linearly cross-dissolving mismatched image regions, which are often due to occlusions or object changes such as opening or closing of eyes over time. In this section, we address these by first computing a per-pixel classification to find those regions. Then, we introduce a perception motivated non-linear blending function to achieve a further improved visual quality of the computed transitions.

6.3.1 Classifying Image Differences

The first step towards the improved blending scheme is to decide on a per-pixel basis if image regions are well matched. Since we already have established correspondence for each pixel with the warping, the problem can be solved by evaluating pixel color differences. Specifically, we use the CIE Lab color space [26] to measure the perceived

per-pixel color difference between the warped images. To relax the influence of differences due to changes in lighting and to focus more on changes due to mismatching we introduce the following difference function:

$$D_{Color}(I_1, I_2) = ||I_1^a - I_1^a||^2 + ||I_1^b - I_1^b||^2 + \gamma \cdot ||I_1^L - I_2^L||^2 \tag{6.6}$$

where I^a and I^b denotes the color channels a and b, I^L is the L channel from Lab space that captures changes in brightness, and γ is a weighting constant. We found a value of 0.25 for γ to give good results in our experiments.

While a simple thresholding can be applied to achieve the classification, this often leads to noisy results since the fact that most neighboring pixels belong to a common region is neglected. Thus, we formulate the problem using the graph-cut approach [17] which allows to take the classification of the neighboring pixels into account. The final mask is then the combination of the classification results of the pixels of I_1 and I_2. An example of a mask derived from the labeling is depicted in Figure 6.5. Note, that the mask has to be computed only once and is warped in parallel to the images during morphing.

6.3.2 Non-linear Image Blending

Based on the classification of image pixel correspondences, we propose to achieve an improved visual quality of the transition with a non-linear adapting blending scheme. Prior image morphing results have proven that differences of matched image regions are well addressed by simple linear cross-dissolve. In the case of object changes and occlusions however, these regions should not be addressed in the same way. A linear cross-dissolve in these areas causes the typical ghosting artifacts depicted in Figure 6.5. The computed transitions between these regions is perceived as getting transparent and vanishing into nothingness by a human observer. Since this is quite unusual, it often draws the observer's attention towards it and thus significantly reduces the plausibility of the transition. On the other hand, it often suffices to produce visual input that is *not* contradicting with the expected visual input on the eye to induce a consistent motion perception in a human observer. From this observation and knowledge about how the human visual system processes motion information (cf. Section 2.2), we propose that presenting *unusual* motion information is worse than presenting non-smooth motion. Specifically, if we use a linear warping scheme combined with a non-linear blending

scheme for mismatching regions, we achieve a significant improvement in the perceived motion plausiblity while obtaining as-smooth-as-possible motion between two static images, cf. Figure 6.5.

To meet this behavior we propose a scaled standard logistic function $b_s(t)$ in the blending step with steepness parameter s

$$b_s(t) = \frac{C_s}{1 + e^{-t\,s}} \tag{6.7}$$

where C_s is a normalization constant dependent on s so that $b_s(0) = 0$ and $b_s(1) = 1$. For $s = 1$, b_s is a simple a linear blending function while for $s \to \infty$ it gradually converges towards the step function with the transition at $t = 0.5$. Using the previously classification mask, matched regions are blended with $s = 1$ and $s > 1$ for pixels in occluding or changing regions.

6.4 Plausible Feature Animation

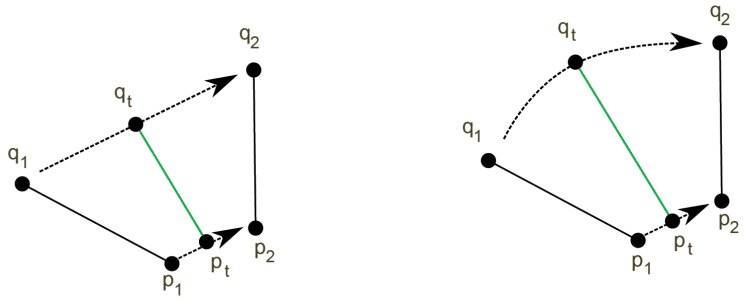


Figure 6.4: Linear interpolation between line feature points causes shrinking and growing of the line during animation (left). As-rigid-as-possible deformation (right) preserve its length during animation and improves the plausibility of the interpolated images.

The work of Beier and Neely [9] proposes to use linear point interpolation on each endpoint of the line features for animation purposes. This however causes unrealistic scalings when the feature rotates between the images, cf. Figure 6.4. Instead, we propose to use an as rigid as possible approach of the line features to interpolate the position of the features during animation. Following [3], we decompose the similarity transformation into its components, namely rotation, translation and scaling. Then we interpolate these components in their respective domain to obtain a plausible feature

animation over time. This avoids the unnatural scaling during rotation and significantly improves the plausibility of the resulting motion, cf. Figure 6.4.

6.5 Motion Layers

The image interpolation covered so far results in a smooth interpolation of the whole image lattice. For some scenes however, motion discontinuities need to be modeled accurately to achieve plausible interpolation results. Especially, for view morphing, background and foreground interpolation usually cannot be faithfully represented by a single image morph. To overcome this limitation, we resort to modeling the different motions by separate layers [154]. First, a segmentation of the foreground and background is obtained by background subtraction techniques or blue screen methods [103]. Then, each layer is morphed separately. The final result is obtained by combining the individual layers via alpha blending in a predefined order. Additional layers can be manually specified if the foreground objects further exhibits motion discontinuities that cannot be addressed by one layer alone [73].

6.6 Implementation

The implementation of our method is divided into a preprocessing and a real-time rendering part. The preprocessing step for image morphing, once line correspondences have been established, is to compute the mask which is used during non-linear blending. This is done by computing the graph cut as described in section 6.3.1 using the implementation of Boykov and Kolmogorov [18].

Rendering is implemented as a multi-pass rendering on graphics hardware. To achieve real-time performance, all steps are implemented as GLSL shaders on the GPU. The most time-consuming operation is hereby the computation of the optical flow. We implemented the algorithm of Horn and Schunck [55] as two fragment shaders, where the first computes the energy term for each pixel of the pre-warped images using the warping defined by the line features. The energy is then successively minimized during ping pong multi pass rendering with the second shader. For the presented results we used a fixed iteration length of 50 iterations to compute the optical flow which showed stable convergence during our experiments. Overall performance of the rendering part

is dependent on the number of line features, the number of layers and the image size. In case of the Capoeira dance sequence with NTSC resolution, rendering the foreground layer with 9 line features takes 2 ms per frame on our test system with a GeForce 7900 GTX and an AMD Athlon64 X2 4800+ Dual Core Processor.

6.7 Results

Our acquisition setup consists of eight off-the-shelf Canon 5D still cameras which feature 12 megapixel resolution and maximally 4 frames per second. The cameras are equipped with 28mm lenses to capture any scene at relatively wide angle. The shutter release on all cameras can be triggered collectively by wire which, however, does not perfectly synchronize shutter release times. For acquisition, the cameras are mounted on tripods which are set up roughly equally spaced around the scene. Neither intrinsic nor extrinsic calibration is performed. When interpolating between cameras we used background subtraction from prerecorded background images to achieve an automatic segmentation of the foreground. We tested our method for space, time, and joint space-time interpolation. In Figure 6.6, we show some example composites of the different space-time interpolations.

6.8 Summary

We have presented an approach for space-time image interpolation based on image morphing of a unsynchronized and uncalibrated set of images. Instead of enforcing physical correctness, our approach is geared towards synthesizing plausible transitions. We showed that the warp field of the Beier and Neely approach can be automatically refined by per-feature followed by per-pixel optimizations to improve its suitability to the task. Then we discussed how occlusions can be addressed by a non-linear image blending scheme taking human perception into account. Finally, we argued that as-rigid-as-possible feature interpolation yields superior results and advocated separate warping of motion layers. Our contributions enable smooth, convincing interpolation across space and time from arbitrary, uncalibrated still images. Conventional, uncalibrated photographs suffice to convincingly interpolate across space, time, and between different objects.

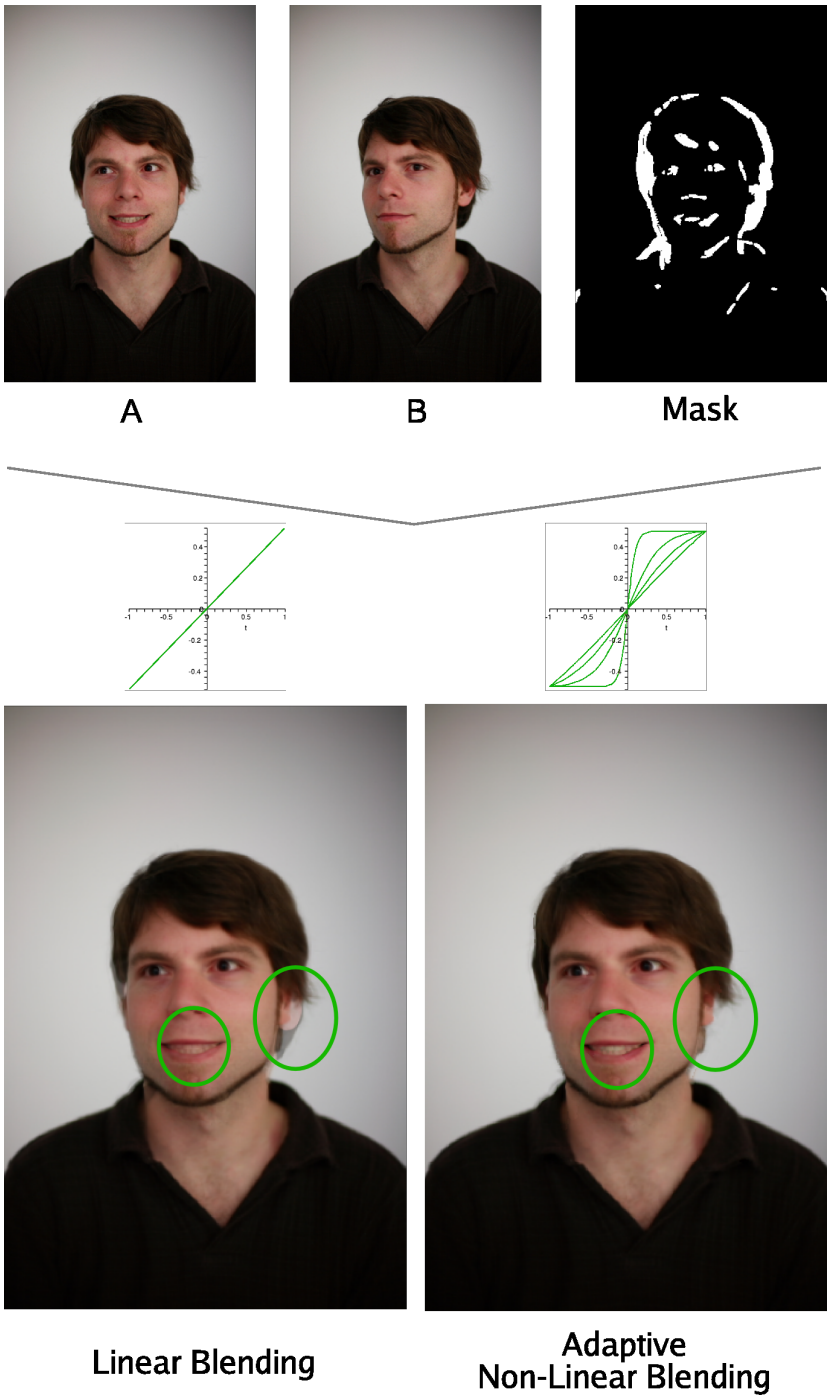


Figure 6.5: The first line shows the two input images and the classification mask computed using graph-cut optimization applied to the perceptual color distance. The bottom line compares between linear blending (bottom left) and our adaptive non-linear image blending method (bottom right) results. (Dis)occluding object regions are blended non-linearly to avoid visually disturbing ghosting artifacts (green circles).

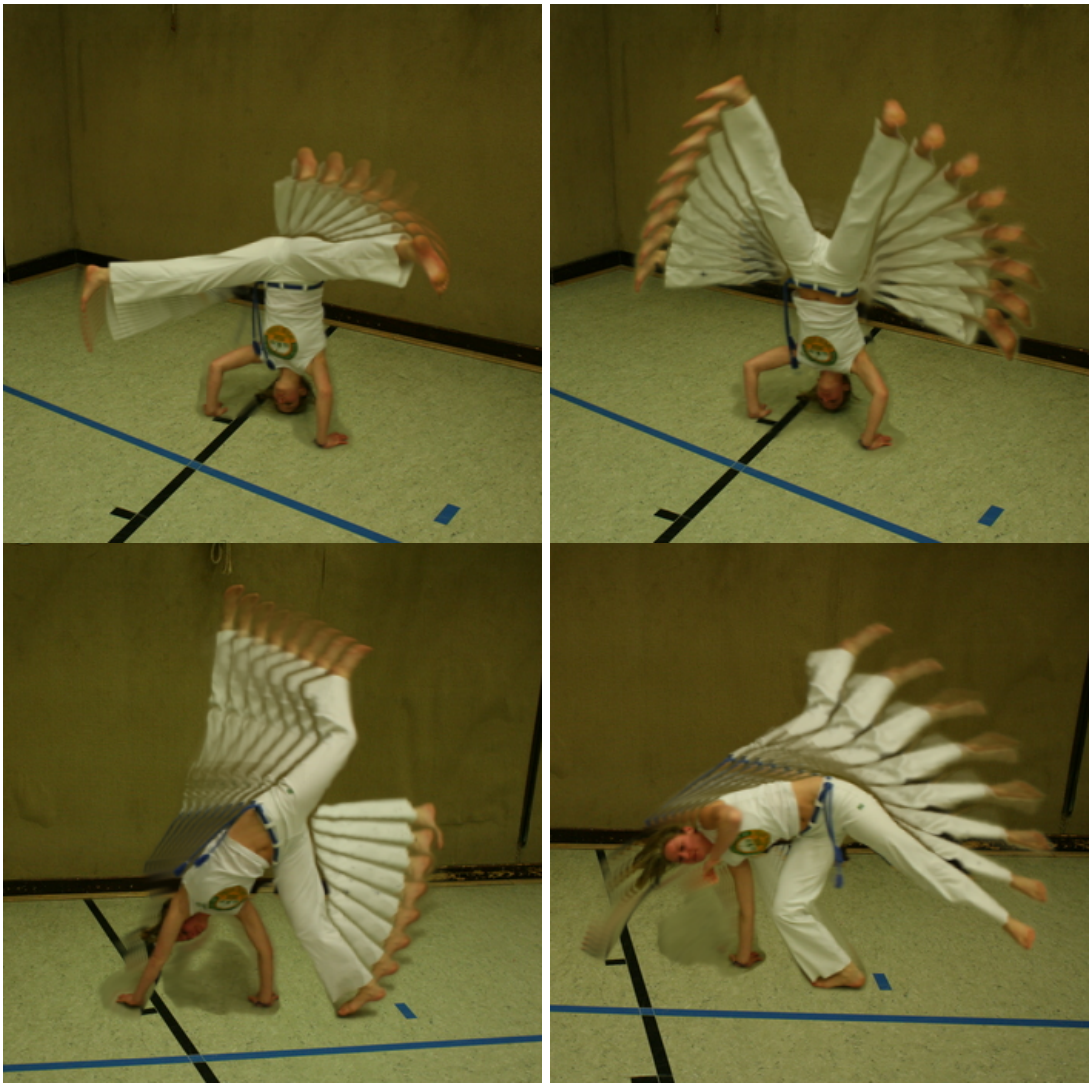


Figure 6.6: Image morphing for visual effects: multi-exposure images created from two consecutive capoeira photos. Several discrete in-between time instants are interpolated and overlaid.

7

Automatic Perception-Aware Space-Time Image Interpolation

7.1 Introduction

In the last section we have introduced a image morphing method based on feature based morphing. While the results achieved are already promising, it is dependent on user input to specifying matching features. On the other hand, automatic image interpolation techniques based on epipolar geometry require to take recordings with synchronized and calibrated cameras. This poses restrictions on their applicability as special camera hardware is necessary and time interpolation is typically not addressed.

In this chapter we propose a novel approach that combines the advantages of both methods. Specifically, we enforce constraints imposed by projective geometry independent of calibration and camera synchronizity to make automatic warping possible. This is achieved by putting forward a novel discontinuity preserving warping method that is based on local homographies between images. By focusing on the features that are specifically important to produce plausible interpolation results, we estimate the parameters of this warping method robustly and quickly. Additionally, occlusions are handled correctly and novel in-between images are rendered without the need to reconstruct 3D surfaces, object motion, camera parameters or segmentation explicitly. With the proposed robust and automatic estimation of the warping, real time space-time interpolation from videos recorded with unsynchronized, uncalibrated cameras becomes possible.

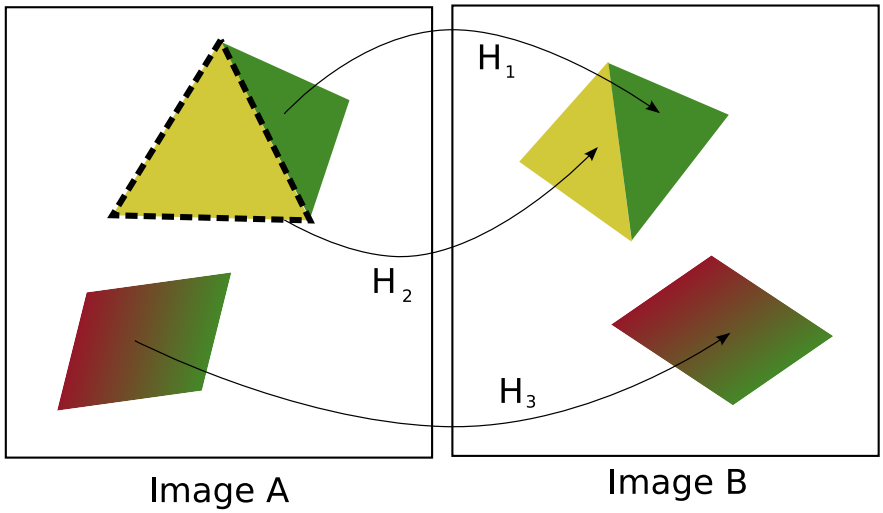


Figure 7.1: Correspondences for views of a dynamic 3D scene consisting of planar surfaces can be described in image space by homographies. We define a translet as the pair of an image segment of a 3D plane and a corresponding homography. For example a translet of image A is the outlined image segment showing the bright face of the pyramid and the corresponding homography H_2 which defines its correspondence to image B.

7.2 A Novel Image Deformation Model for Time and View Interpolation

The relation between two projections of a 3D plane can be directly described via a homography in image space (cf. Section 2.3.1). Such homographies for example describe the relation between a 3D plane seen from two different cameras, the 3D rigid motion of a plane between two points in time seen from a single camera or a combination of both. Thus, the interpolation between images depicting a dynamic 3D plane can be achieved by a per pixel deformation according to the homography directly in image space without the need to reconstruct the underlying 3D plane, motion and camera parameters explicitly (cf. Figure 7.1). The relation between the corresponding pixels of images from a typical dynamic real-world scene on the other hand is of course far more complex. However, many graphics approaches have been very successful in creating photo-realistic images from approximations of natural scenes and objects with meshes consisting of simple planar triangles. For each such triangles the relation of the corresponding pixels is again exactly described via local homographies.

Our proposed image deformation model is motivated by these observations. We assume that natural images can be decomposed into regions, for which the deformation of each element is sufficiently well described by a homography. Specifically, we introduce *translets* which are homographies that are spatially restricted. That is, a translet is described by a 3×3 matrix H and an image segment. To obtain a dense deformation, we enforce that the set of all translets is a complete partitioning of the image and thus each pixel is part of exactly one translet. Note, that since the deformation model is defined piecewise, it can well describe motion discontinuities as for example resulting from occlusions.

7.3 Estimating the Image Deformation

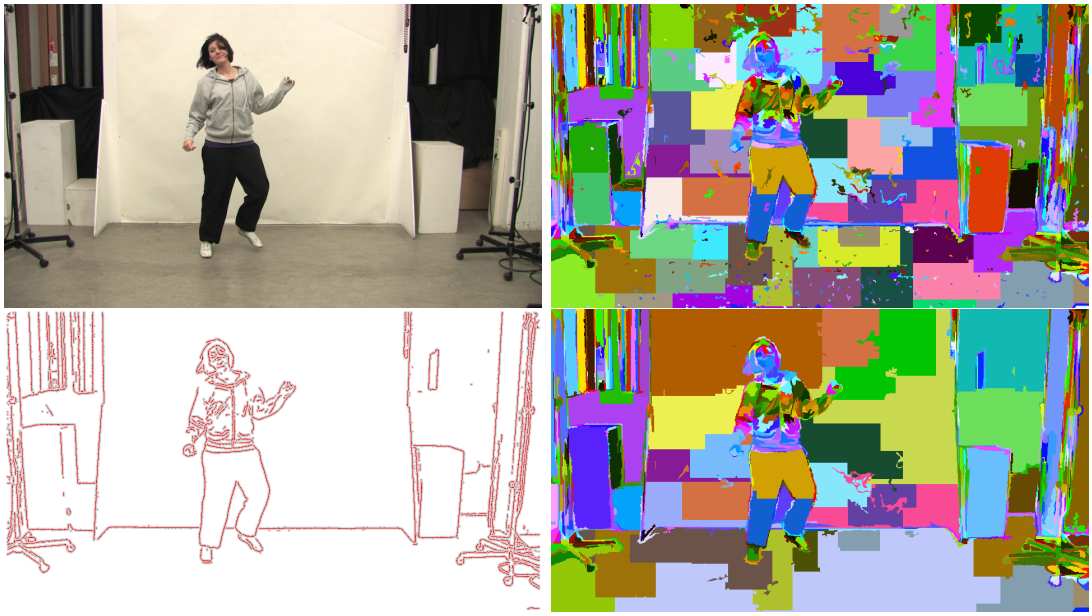


Figure 7.2: An image (upper left) and its decomposition into its homogeneous regions (upper right). Since the transformation estimation is based on the matched edglets, only superpixels that contain actual edglets (lower left) are of interest. We merge superpixels with insufficient edglets with their neighbors (lower right).

In this section we discuss how to robustly estimate dense correspondences between two images based on the proposed model. Therefore, both a suitable partitioning of the images into regions that can be approximated by 3D planes and sparse point correspon-

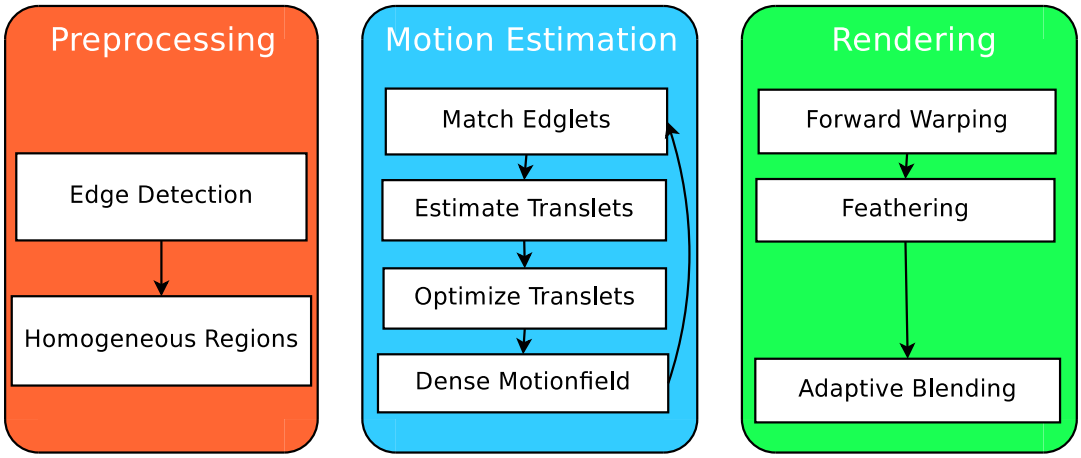


Figure 7.3: Overview of the proposed image interpolation approach: first, the images are preprocessed to find edges and homogeneous regions. These are then used to determine dense correspondences. Finally, we use this correspondence field for interpolation rendering of image transitions in real-time.

dences have to be established. Then, homographies from the point correspondences are estimated for each region to form a translet. While the optimal partitioning of the images into such regions is not known a priori, this has great influence on the solution. A small number of regions will result in a very robust but restrictive solution, while a larger number increases the flexibility at the cost of decreased robustness against outliers in the match. To obtain an optimal result, we follow a bottom up approach. We start from a large number of regions and merge neighboring translets in a greedy manner until the optimal ratio is achieved. In the following, we discuss the steps in estimating the image deformation between two images in detail. See also Figure 7.3 for an overview of the proposed image interpolation approach.

7.3.1 Matching of Edge Pixels

The first step in estimating the parameters of the deformation model is to find a set of point correspondences between the images. These will be used in the second step to estimate the homographies. Additionally, for a plausible solution in terms of perceived motion quality, it is also necessary that these point features are also considered important for human vision. Edges and corners are thus especially suited point feature candidates. They are both relatively stable over time and viewpoints, are the image

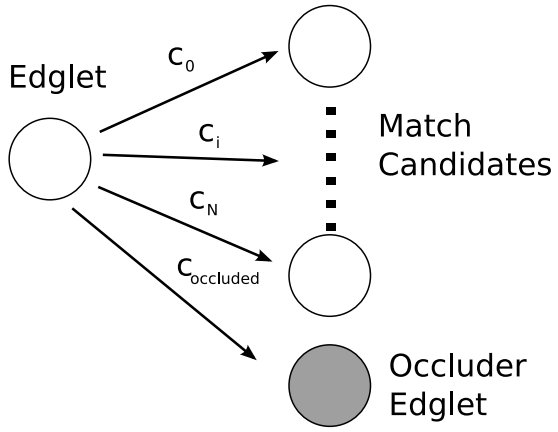


Figure 7.4: Subgraph of the weighted bipartite graph matching problem for a single edglet. Each edglet has an edge to its possible match candidates and an additional edge to its virtual occluder edglet.

parts where motion is most apparent and also the features that the human visual system is known to measure very early, cf. Section 2.2. For the detection of edge pixels, we can also resort to a large body of previous work. Specifically, we used the Compass operator [117] in our experiments as it has the advantage to directly make use of color information and often outperforms the Canny operator [21]. After non-maximal suppression, we obtain a set of edge pixels or *edglets*, cf. Figure 7.2. Depending on the scene, between 2000 to 20000 pixels are edglets (cf. Figure 7.2).

To establish a good match between the edglets of the two images to interpolate, it is necessary to match them as complete as possible and to consider the spatial context of each edglet to preserve local structure. We will ensure completeness by posing the matching problem accordingly and address local structure preservation with a stable descriptor that captures spatial context. The shape context descriptor [10] has been shown to perform very well at capturing the spatial context of the nearest k neighbor edglets and is robust against the expected deformations. Completeness of the matching of the edglets based on euclidean distance and shape context is then achieved by solving an maximum weighted bipartite graph matching problem. The additional advantage is that this problem can be solved globally optimal in the matter of seconds for the problem sizes we are facing [11]. One prerequisite for the reformulation is that for each edglet in the first set a match in the second set exists, otherwise the completeness cannot be achieved. While this is true for most edglets, some will not have a correspondence in

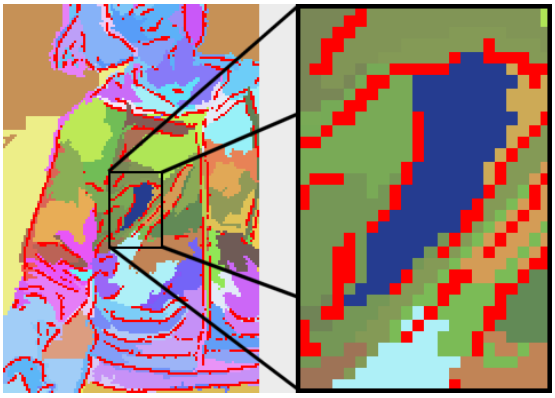


Figure 7.5: The translets of an image are found by partitioning the image according to a superpixel segmentation and computing local homographies from point correspondences to the target image.

the other set due to occlusion or small instabilities of the edge detector at faint edges. However, this is easily addressed by inserting virtual occluder edglets for each edglet in the first edglet set. The graph for the matching problem is then build as depicted in Figure 7.4. Each edge pixel of the first image is connected by a weighted edge to its possibly corresponding edge pixels in the second image and additionally to its virtual occluder edglet. The weight or cost function edglet $\mathbf{e}_i$ in I_1 and $\mathbf{e}'_j$ in I_2 is then defined as

$$C(\mathbf{e_i}, \mathbf{e}'_j) = C_{dist} + C_{shape} \tag{7.1}$$

where the cost for the shape is the χ^2-test between the two shape contexts and the cost for the distance is defined as

$$C_{dist}(\mathbf{e_i}, \mathbf{e}'_j) = \frac{a}{(1 + e^{-b\,||\mathbf{e}_i - \mathbf{e}'_j||})} \tag{7.2}$$

with $a, b > 0$ such that the maximal cost for the euclidean distance is limited by a. The cost $C_{occluded}$ is user defined and controls how aggressively the algorithm tries to find a match with an edglet of the second image. The lower $C_{occluded}$ the more conservative the resulting matching will be, as more edges will be matched to their virtual occluder edglets.

7.3.2 Estimating the Local Homographies

According to the proposed motion model, we assume that the scene can be approximated with piecewise planes for interpolation purposes. For each such region, we would assume that the motion is described by the relation of projections of a 3D plane as discussed in Section 7.2. From Gestalt theory [156] it is known that for natural scenes, these regions share not only a common motion but in general also share other properties such as similar color and texture, cf. Section 2.2. Felzenszwalb and Huttenlocher [42] proposed to partition images into so called superpixels based on neighboring pixel similarities, cf. Figure 7.2. We resort to their algorithm to find an initial partitioning of the image into regions to become translets. Then from the matching between the edge pixels of the images to interpolate, local homographies for each set of edge pixels in the source image that are within one superpixel are estimated, cf. Figure 7.5. Since the least-squares estimation based on all matched edglets of a translet is sensitive to outliers and often more than the minimal number of four matched edge pixels is available, a RANSAC approach to obtain a robust solution and filter match outliers is preferred instead [52].

7.3.3 Translet Optimization

From the point correspondences we have established dense correspondences between the images using our deformation model. However, in our experiments we observed that between 20% to 40% of the computed matches are outliers and thus some translets will have wrongly estimated transformations. We address this problem by optimizing the number of translets of our image deformation model to increase the robustness against these outliers. The initial solution of our model is generally very flexible and suffers from numerical instability, because the spatial support of the translets can be too small for a reliable estimation. Using a greedy approach, we iteratively merge the most similar transformed neighboring translets into one, as depicted in Figure 7.6, until the ratio of outliers to inliers is lower than a user defined threshold. When two translets are merged, the resulting translet then contains both edglet sets and has the combined spatial support. The homographies are re-estimated based on the new edglet set and the influence of the outliers is again reduced by the RANSAC filtering.

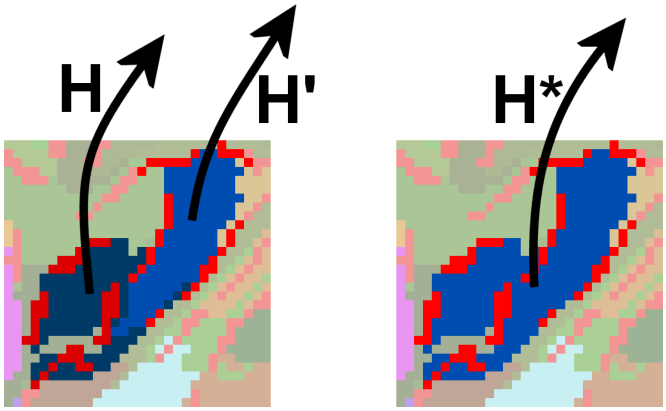


Figure 7.6: During optimization, similar transformed neighboring translets are merged into a single translet. After merging, the resulting translet consists of the combined spatial support of both initial translets (light blue and dark blue) and their edglets (light red and dark red).

7.3.4 Per-Pixel Correspondences

So far motions and discontinuity are handled on the translet level. However, when only a part of a translet boundary is at a true motion discontinuity, noticeably incorrect discontinuities still produce artifacts along the rest of the boundary. For example, the motion of an arm in front of the body is discontinuities along the silhouette of the arm, while the motion at the shoulder changes continuously. Additionally, small deviations from the planar motion are not sufficiently well handled by the general approach. Thus we address these issues on a per-pixel basis. Since the translets partition the image, each pixel in the image is uniquely associated with a translet t. The deformation vector for a pixel $\mathbf{x}$ is thus computed as

$$d(\mathbf{x}) = H_t \cdot \mathbf{x} - \mathbf{x}. \tag{7.3}$$

We can then resolve the per pixel smoothing by an anisotropic diffusion [102] on this vector field using the diffusion equation

$$\delta I/dt = div(\ g(min(|\nabla d|, |\nabla I|)\ \nabla I) \tag{7.4}$$

which is dependent on the image gradient ∇I and the gradient of the deformation vector field ∇d whichever is smaller in magnitude at the observed pixel. The function g is a simple mapping function as defined in [102]. Thus, the deformation vector field

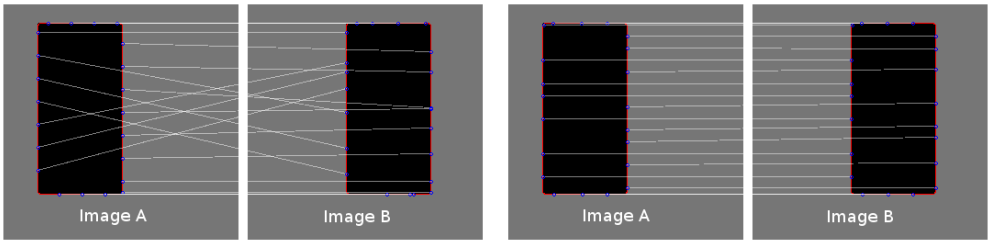


Figure 7.7: Local matching minima (left) can be avoided by multiple iterations. In a coarse to fine manner, in each iterations the number of translets increases avoiding local matching minima by using the previous result as prior (right).

is smoothed in regions that have similar color or similar deformation, while discontinuities that are both present in the color image and the vector field are preserved. This improves the smoothness of the deformations on a per-pixel level while preserving important motion discontinuities. During the anisotropic diffusion, edglets that have an inlier match, meaning they are in only slightly deviating from the planar model, are considered as boundary conditions of the PDE. This results in exact edge transformations handling also non-linear deformations for each translet and significantly improves the achieved quality. The total timings for the computation of the deformation field for different resolutions and scenes are listed in Table 7.1

7.3.5 Multiple Iterations and User Interaction

Since our matching energy function (Eq. 7.1) is based on spatial proximity and local geometric similarity, we can introduce a motion prior by pre-warping the edglets with a given deformation field. The estimated dense correspondences described in the last sections can be used as such a prior. We can then implement a coarse to fine iterative approach to overcome local matching minima, as for example depicted in Figure 7.7, as follows: In the first iteration, we optimize the number of translets until we obtain the coarsest possible deformation model with only one translet and thus approximate the underlying motion by a single perspective transformation. During consecutive iterations, the threshold is decreased to allow for more accurate deformations as the number of final translets increases. Using the previous solution as motion prior significantly reduces the risk to getting stuck in local matching minima, cf. Figure 7.7.

Additionally, solving on different image resolutions similar to scale-space [168] further improves robustness. Especially in cluttered images, the obtained solution significantly profits from first solving on small scale and upsampling the solution to the next level as prior, cf. Figure 7.8.

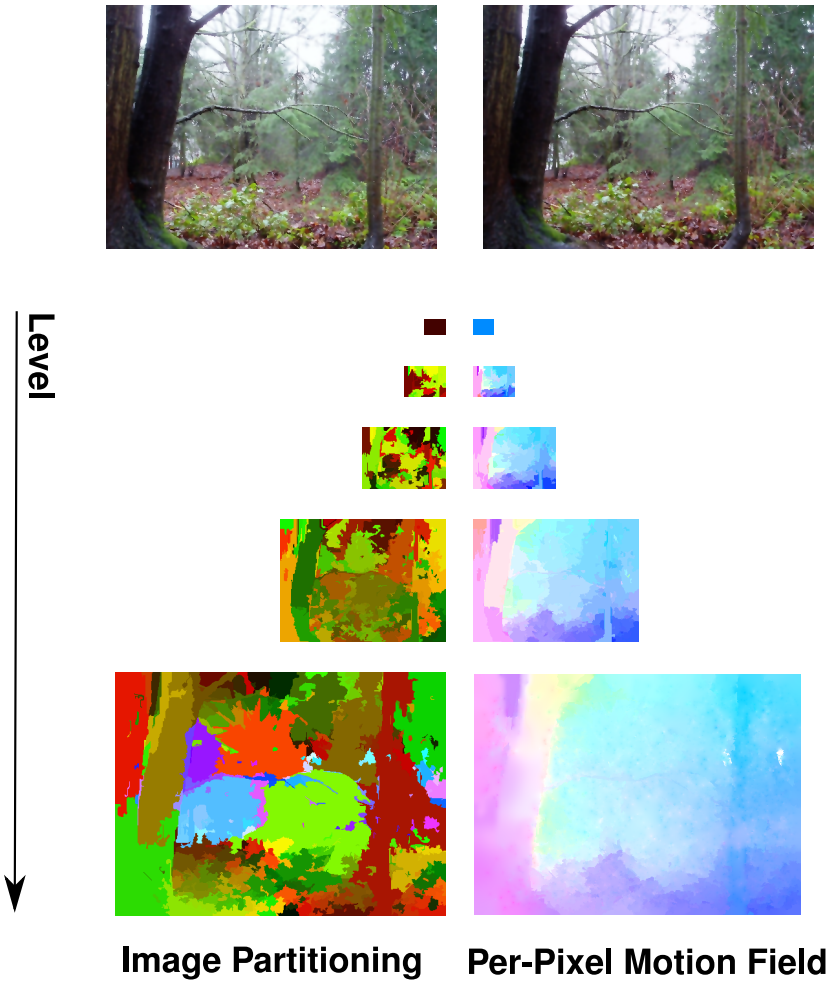


Figure 7.8: First solving on small image scales and upsampling of the found solution significantly improves the quality especially for cluttered scenes. Input images are courtesy of Larry Zitnick

In rare cases, some scenes still can not be matched automatically sufficiently well. For example when similar structures appear multiple times in the images the matching can get ambiguous and only be addressed by high level reasoning. To resolve this, regions can be selected in both images by the user and the automatic matching is

computed again only for the so selected subset of edglets. Due to this restriction of the matching, the correct match is found and used to correct the solution.

Table 7.1: Timing results of our method on a AMD Athlon(tm) 64 X2 Dual Core Processor 4800+, 4GB RAM, NVIDIA GeForce 7800 GTX to compute the dense correspondences. If users interact to improve the solution only parts are recomputed which reduces the response times.

Scene	Edglets	Res.	Matching	Optim.
Dancer	2570	960x540	1.94 s	5.67 s
Dimet.	8604	584x388	11.27 s	16.54 s
Rub.Whale	13474	584x388	19.04 s	29.87 s
Hair	17560	960x540	27.77 s	35.57 s

7.4 Interpolation Rendering

Rendering in-between images is achieved by applying the correspondence field estimated with our image deformation model to the images and blending these warped images. This can be implemented on graphics hardware using per-vertex mesh deformation and alpha blending with real-time rendering performance. To get the deformations for the in-between images we linearly interpolate the deformation vector field.

7.4.1 Warping with Occlusions

We implemented the forward warping by a per-vertex deformation of a regular planar triangle mesh of the image plane, where each pixel in the image is represented by a quad with appropriate texture coordinates. Two problems arise with forward warping at motion discontinuities: Fold-overs and missing regions.

Fold-overs occur when two or more pixels in the image end up in the same position during warping. This is the case when the foreground occludes parts of the background. Consistent with motion parallax we assume that the faster moving pixel is closer to the viewpoint to resolve this conflict. When on the other hand regions get dis-occluded during warping the information of these regions is missing in the image and must be filled in from the other image. This leaves two options in this case: cutting the mesh at the motion discontinuities *before* warping or detecting triangles that span over these

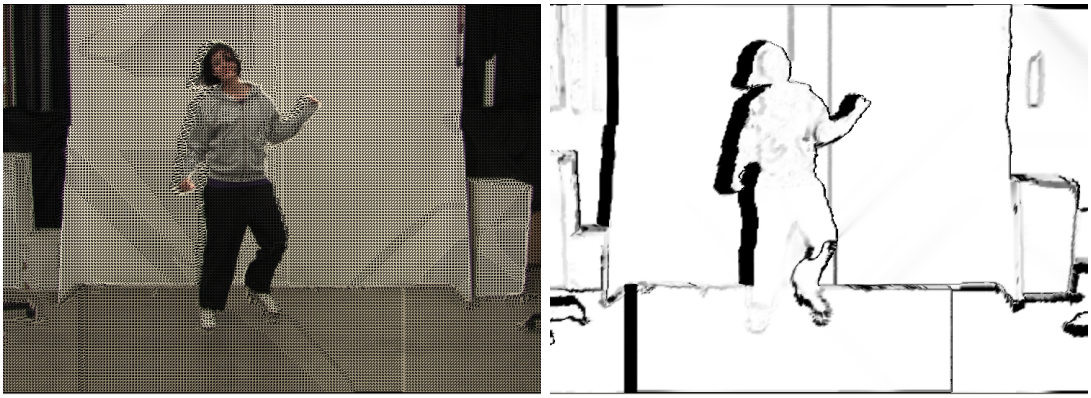


Figure 7.9: Left: Per-vertex mesh deformation is used to compute the forward warping of the image, where each pixel corresponds to a vertex in the mesh. The depicted mesh is at a coarser resolution for visualization purposes. Right: The connectedness of each pixel that is used during blending to avoid a possibly incorrect influence of missing regions.

discontinuities *after* rendering. Mark et al. [80] pointed out that the second approach performs better and proposed a connectedness criterion evaluated on a per-pixel basis after warping. We adapt this measure and compute it directly from the divergence of the deformation vector field such that

$$c_A = 1 - div(d_{AB})^2. \tag{7.5}$$

with c_a the connectedness and d_{AB} vector field between the images A and B (cf. Figure 7.9). The connectedness is computed on the GPU during blending to adaptively reduce the alpha values of pixels with low connectedness. Thus, in missing regions only the image which has the local information has an influence on the rendering result.

7.4.2 Feathering

At fold-overs, the warped images can have jaggy artifacts due to aliasing problems of the rendering. Opposed to recordings with cameras, rendered pixels at the boundaries are not a mixture of background and foreground color but are either foreground or background color. However, these artifacts occur only at large motion discontinuities, which can be robustly discriminated by the local change in the motion vectors by simple thresholding (cf. Figure 7.10). In a second rendering pass, we model the color mixing of foreground and background at boundaries using a small selective low-pass filter applied

only to the detected motion boundary pixels. This effectively removes the artifacts with a minimal impact on rendering speed and without affecting rendering quality in the non-discontinuous regions.

7.4.3 Multiple Image Interpolation

We can describe the interpolation between two images A and B as

$$I(\alpha) = \frac{c_A(1-\alpha) \cdot [A \circ d_{AB}(\alpha)] + c_B(\alpha) \cdot [B \circ d_{BA} \cdot (1-\alpha)]}{c_A(1-\alpha) + c_B(\alpha)} \tag{7.6}$$

where $c_X(\phi)$ is the locally varying influence of each image on the final result which is modulated by the connectedness

$$c_A(\alpha) = c_A \cdot \alpha \tag{7.7}$$

Thus, the (possibly incorrect) influence of pixels with low connectedness on the final result is reduced.

The interpolation is not restricted to two images. Interpolating between multiple images is achieved by iteratively repeating the warping and blending as described in (7.6), where I takes over the role of one of the warped images in the equation. To stay inside the image manifold that is spanned by the images the interpolation factors must sum to one, $\sum_i \alpha_i = 1$. We elaborate more on this topic in Chapter 10.

7.5 Results

First, we compared our results to interpolation results based on state-of-the-art optical flow methods using the Middlebury examples [6] (cf. Table 7.2, Figure 7.11). Since these methods do not allow for user interaction we compare the results of our unimproved automatic results. As can be seen our approach is best when looking at the interpolation errors and best or up to par in the sense of the normalized interpolation error. We also like to point out that from a perception point of view the normalized error is less expressive than the unnormalized error since discrepancies at edges in the image (e.g. large gradients) are dampened. Interestingly, relatively large angular errors are observed with our method emphasizing that the requirements of optical flow estimation and image interpolation are different.

Figure 7.10: Jaggy artifacts due to aliasing artifacts can get visible at motion discontinuities. These are however easily discriminated by a threshold on the motion field. In a second rendering pass we correct the previously detected artifacts.

In addition, we recorded dynamic scenes with conventional, unsynchronized, and uncalibrated video cameras. Some results are depicted in Figure 7.12. We used off the shelf Canon HDV camcorders that have a horizontal field of view of $\approx 50°$ and were spaced apart by $\approx 15°$. The shown scenes also contain surfaces that are hard to reconstruct in 3D and are thus problematic for typical image based rendering methods, such as the flame of the fire-breather and the flying hair of the woman.

7.6 Summary

In this chapter, we have presented a novel interpolation method for view and time interpolation directly in image space. We introduced our image deformation model that is used to enforce relaxed physical constraints on the estimation of dense correspondences based on homographies between planes in 3D space. Favorable properties of this model are that edges are transformed exact and that motion discontinuities are preserved, and thus occlusions are handled appropriately. The benefit of our approach is that we can robustly estimate correspondences between images recorded with unsynchronized and uncalibrated cameras. We compared the results with other general state-of-the art interpolation methods and showed that our method performs very well in terms of interpolation error on a set of standard examples.

Table 7.2: Interpolation, Normalized Interpolation and Angular errors computed on the Middlebury Optical Flow examples by comparison to ground truth with results obtained by our method and by other methods taken from [6].

Venus	Interp.	Norm. Interp.	Ang.
Our Method	**2.88**	**0.55**	**16.24**
Pyramid LK	3.67	0.64	14.61
Bruhn et al.	3.73	0.63	8.73
Black and Anandan	3.93	0.64	7.64
Mediaplayer	4.54	0.74	15.48
Zitnick et al.	5.33	0.76	11.42

Dimetrodon	Interp.	Norm. Interp.	Ang.
Our Method	**1.78**	**0.62**	**26.36**
Pyramid LK	2.49	0.62	10.27
Bruhn et al.	2.59	0.63	10.99
Black and Anandan	2.56	0.62	9.26
Mediaplayer	2.68	0.63	15.82
Zitnick et al.	3.06	0.67	30.10

Hydrangea	Interp.	Norm. Interp.	Ang.
Our Method	**2.57**	**0.48**	**12.39**

RubberWhale	Interp.	Norm. Interp.	Ang.
Our Method	**1.59**	**0.40**	**23.58**

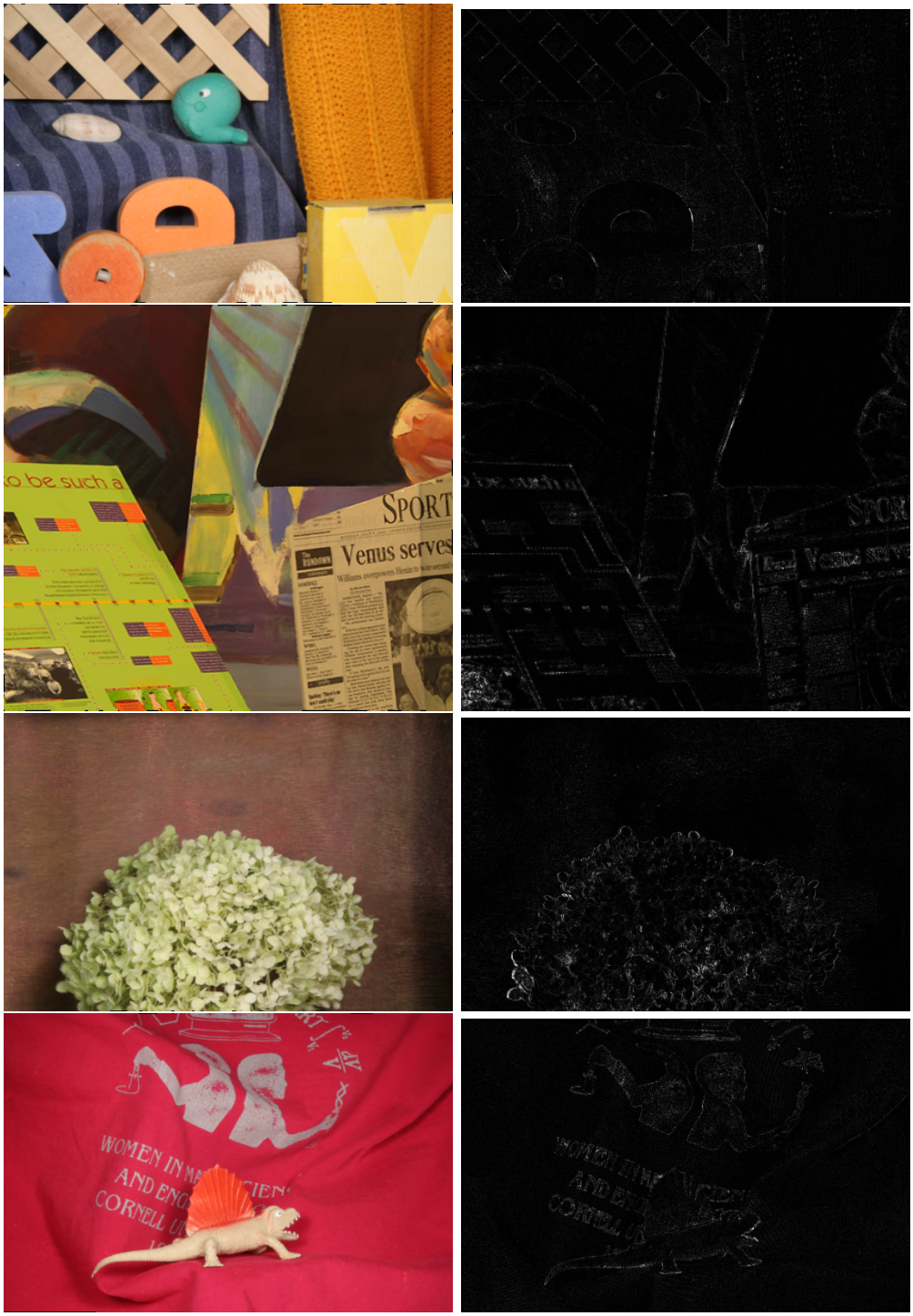


Figure 7.11: Results on the Middlebury dataset [6]. (Left Column) In-between image automatically computed with our method. (Right Column) Contrast-stretched difference to ground truth.

Key frame | Temporal motion blur effect | Key frame

(a)

Key frame | Progressive motion distortion effect | Key frame

(b)

Key frame | Local motion compensation effect | Key frame

(c)

Figure 7.12: Different visual effects created using the presented image interpolation method.

7. AUTOMATIC PERCEPTION-AWARE SPACE-TIME IMAGE INTERPOLATION

8

A Psychophysical User-Study on Image Interpolation

8.1 Introduction

In this chapter we evaluate the image interpolation algorithm introduced in Chapter 7 in terms of parallels to processes of the human visual system to understand seen images. In a user study we confirm the perceptual validity of each part of the image interpolation algorithm. We quantify changes in perceptual quality introduced by parameter changes within our proposed approach, compare the results against other approaches and investigate whether there is a perceptual difference between results on real-world and synthetic image material.

8.2 Perceptual Criteria for Image Interpolation

Human vision is a very powerful system, adept at extracting meaningful patterns so that we can understand, navigate through, and interact with our surroundings rapidly and efficiently. The importance and complexity of this task is perhaps reflected by the fact that approximately half of our brain is dedicated to processing visual input. While the human visual system as a whole is very complex and has many not yet fully understood aspects, some parts are well researched, see also Chapter 2.2

Based on his work with flies and beetles, [111] mathematically and neurally described a local-correlator motion detector. The detector, which explicitly relies on the fact that real-world objects tend to move rather smoothly, matches small image patches

across small spatial and temporal distances. Interestingly, low-level motion processing in humans is also well described by this detector [54, 107]. Another finding is that the human visual system seems to take advantage of the fact that neighboring areas on an object tend to have the same motion. Local smoothness constraints help to compensate for noise and aid in image segmentation. Additionally, the common motion of neighboring patches and differently oriented edges are used to help solve the aperture problem [151] .

From these findings on the human motion perception we conclude that to achieve perceptually plausible image interpolation results, it is important to transform corresponding edges exactly onto each other while transforming homogeneous regions within the images coherently. The algorithm introduced in Chapter 7 can also be interpreted in terms of parallels to the motion processing of the human visual system. Actually, the different steps in the derivation of the motion field improve the quality of the interpolation in a perceptually plausible sense. The matching of the edglets ensures that edges are transformed exactly onto each other. Then the estimation of local transformations and the final per-pixel smoothing of the motion field indeed optimizes the coherence of the motion field. To evaluate this interpretation in a more concise way, we conduced a user study.

8.3 User Study

In order to assess the perceptual quality of our interpolation algorithm, we ran a psychophysical validation study which had three major goals. First to quantify changes in perceptual quality introduced by parameter changes *within* our proposed approach to image interpolation. Second to compare the results of the proposed algorithm against other approaches to image interpolation. And third to investigate whether there would be a perceptual difference between results on real-world and synthetic image material.

8.3.1 Stimuli

Guided by our goals, we selected a total of nine different approaches for creating interpolated image sequences. The input consisted of several sequences depicting rotations around objects. From these we kept every third frame and used the algorithms for interpolating the missing two intermediate frames. For the used scenes this was the

largest gap that the automatic approaches could interpolate with reasonable quality - note also, that this corresponds to changes in viewing angle of around 10 degrees on average. The surfaces are dominantly diffuse since specularities must be treated as transparent entities to model their motion correctly. The following list describes the algorithms in more detail:

original: as the baseline, we compared all algorithms against the original video sequence showing the full, smooth motion

blend: a simple blending algorithm which creates intermediate frames by blending between two consecutive key-frames

opticalflow: a physically motivated optical-flow algorithm [55] was used to compute the motion field for the interpolation

nooptim: our automatically computed initial transformation solution after the second iteration without further optimization of the translets

optim100: the result of our method including translet optimization but without per-pixel diffusion after the second iteration (cf. section 7.3.2)

nofeathering: the result of our method but without the feathering at motion discontinuities during rendering (cf. section 7.4.2)

firstit: the result of our method after the first iteration with optimization of the correspondence field and subsequent diffusion

full: the result of our proposed automatic perception-based image interpolation algorithm after the second iteration with optimization of the correspondence field and diffusion

corrected: the result of our automatic approach with additionally manually corrected local errors as discussed in section 7.3.1

The first three conditions together with the *full, corrected* conditions address the goal of comparing different approaches to interpolation, whereas conditions *firstit, nooptim, optim100, nofeathering* were designed to compare the perceptual quality of different parameter settings.

In order to address our third goal of comparing performance differences of the algorithm on real-world and synthetic images, we used the two different types of scenes shown in Figure 8.1. Four scenes showed computer-generated sequences of objects rotating around the vertical axis for 180 degrees. The two real-world scenes showed a plant and books which were recorded with a digital video camera.

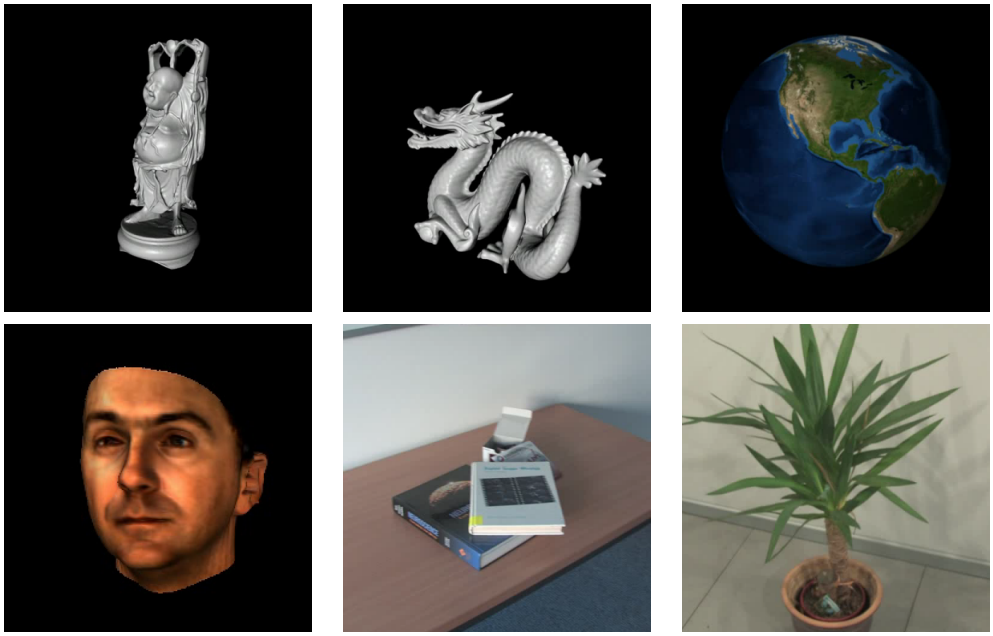


Figure 8.1: The six different scenes used in the psychophysical validation study. The first four scenes consist of computer-generated 3D objects, whereas the fifth and the sixth scene were recorded indoors with a standard hand-held camera.

8.3.2 Experimental design

Rather than using a standard rating task in which participants would be shown a sequence and be asked to rate its quality, we opted for a more systematic approach. In the psychophysical study, we used a two-alternative-forced-choice task in which two video sequences were shown successively and participants were asked to indicate which sequence contained more visual artifacts. Such a direct comparison allows for a more fine-grained analysis of the data as rating tasks are often subject to scaling problems [152]. For each of the 6 different scenes we compared all 9 different interpolation algorithms against each other (only doing AB and AA, not BA comparisons), yielding a total of $6 \cdot (9 \cdot \frac{8}{2} + 9) = 270$ trials.

All scenes were rendered at 500x500 pixels with 25 frames per second and were 3-5 seconds long. Sequences were presented on a black background on a CRT monitor using a pixel resolution of 1024x768 at 75Hz. Participants viewed the stimuli at a distance of roughly 50cm while sitting in a dark room. Each trial consisted of a fixation cross shown for 1 second, followed by the first sequence, a second fixation cross for 0.5 seconds, and

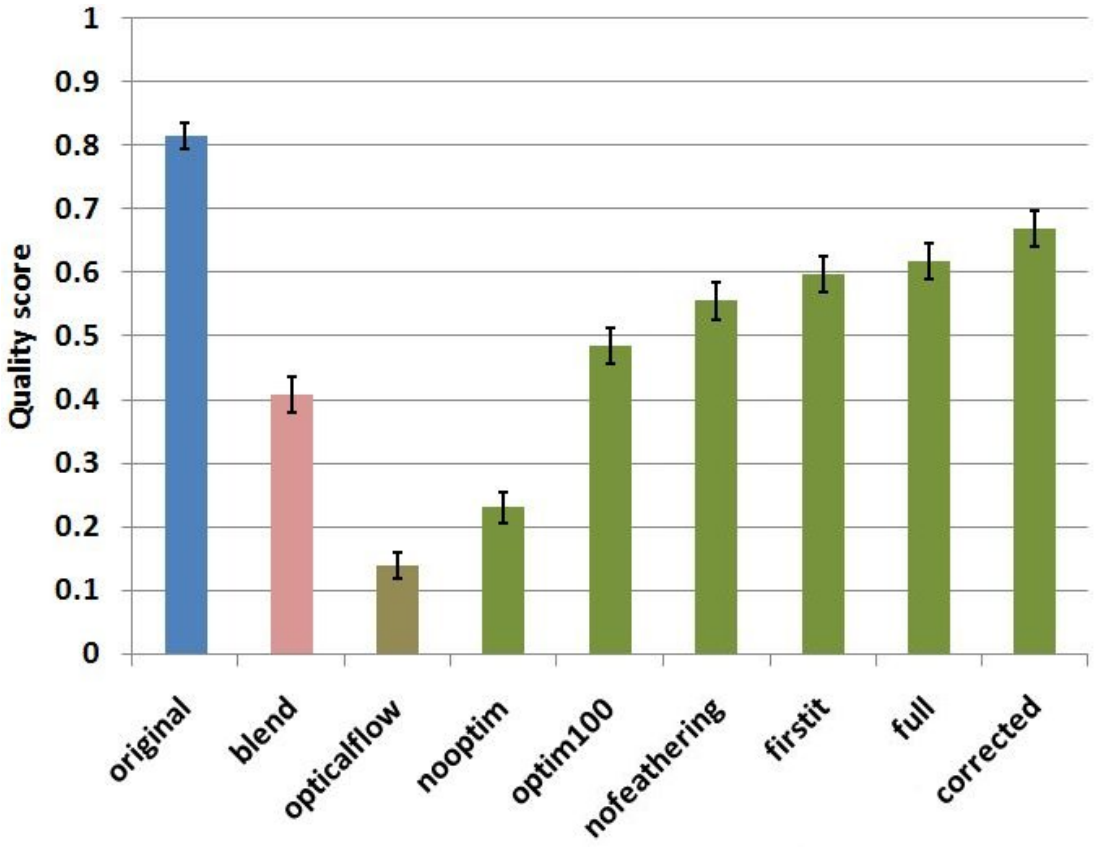


Figure 8.2: Perceptual quality scores for nine different test conditions (image interpolation schemes).

the second sequence. After this, the screen was blanked and participants were asked to indicate by a key press which sequence contained more visual artifacts. Participants were briefed before the experiment that in this case artifacts were defined as "any visual disturbances resulting in non-smooth motion". All participants completed three test trials before the experiment, which were used to get them acquainted with the task. Neither during the test trials nor during the experiment was any feedback given and none of the participants reported any difficulty with doing the task. The whole experiment lasted around 90 minutes. Our test group consisted of 10 participants who did *not* have any graphics-related background.

8.3.3 Analysis

For the first analysis, we determined a perceptual quality score for each algorithm by counting how many times it was chosen as producing fewer visual artifacts when compared to one of the other algorithms. The normalized scores are shown in Figure 8.2 for all nine approaches. The following analysis addresses our first two experimental questions, by interpreting the results for each image interpolation approach (all statistical tests were run as one-tailed t-tests corrected for multiple comparisons).

original: The original sequences are rated as having the best perceptual quality (all $p<0.01$)[1].

blend: Despite the technical simplicity of this condition, the quality score is still reasonably high. Whereas this might be surprising at first glance, the perceptual impression of the resulting motion is that of a jerky, but very consistent motion.

opticalflow: The interpolation results in this condition are rated as having the worst quality (all $p<0.01$). This seems to be due to the fact that the scenes contain motion discontinuities which are not adequately handled by the Horn-Schunck approach [55] and cause local instabilities in the computed motion field. This violation of the object contour stability has a negative impact on the perceptual quality of the sequences.

nooptim: Within the parameter changes of our approach this is the worst condition (all $p<0.01$). As the motion field in this condition is computed from only the matches, this often resulted in sharp spikes and discontinuity errors. This finding underlines the importance of increasing the motion coherence during the optimization (cf. section 7.3.2). As discussed in section 8.2 motion coherence is one of the most crucial properties to achieve a perceptually plausible image interpolation.

optim100: Compared to the *nooptim* condition, the increase in perceptual quality due to the optimization of the correspondence field is dramatically demonstrating the importance of producing locally consistent motion for perceptual fidelity.

nofeathering: According to our expectations the feathering of edges for our scenes has a perceptually noticeable positive effect as unnatural strong color gradients produced by the rendering as discussed in section 7.4.2 are removed.

[1] The p-value is the probability of obtaining a result at least as extreme as the one that was actually observed, given that the null hypothesis is true. This means, for a given significance level, typically 0.05 or even stronger 0.01, the comparison with the p-value decides if the null hypothesis can be safely rejected.

firstit: The results show that already after the first iteration, the estimated motion field computed with our method produces perceptually pleasing motion as reflected in the high quality scores. Compared to the *optim100* condition, there is another significant increase in perceptual quality. This increase is due to the per-pixel anisotropic diffusion, which further improves the motion consistency of the results while preserving the discontinuities.
full: The perceptual quality shows that on average the improvement of several iterations lies within the measurable accuracy.
corrected: Not surprisingly, of all the approaches based on the proposed pipeline this condition fared best (all $p<0.05$). The difference between this and the full condition is small but significant showing that a small amount of user interaction can improve the results further (cf. section 7.3.1).

In order to address the third experimental question of quality differences between real-world and synthetic scenes we compared how many times participants chose the *corrected* over the *original* condition. As Figure 8.3 shows, for both real-world scenes, only one response was given in favor of the corrected scene, whereas for the face sequence it seems that participants could not decide which of the two conditions was better, as preference was at 50%.

Taken together, these results have shown that our proposed approach to image interpolation already produces perceptually plausible, high-quality interpolations. Whereas there is still some room for improvement - especially for identifying invalid correspondences and improving robustness against outliers - the quality of the sequences is surprisingly good given that no prior knowledge about camera calibrations, scene geometry, or object identity was used. Additionally, the results confirm and extend the perceptual approach to computer graphics - that our visual system has evolved to deal with natural *image statistics* (things tend to move smoothly; objects have well-defined, stable boundaries, etc.) rather than to explicitly and accurately reconstruct the 3D world from visual input (simple image morphing can be enough).

8.4 Summary

In this chapter, we discussed how the interpolation method introduced in Chapter 7 can be related to human motion perception. It has several advantages over physically

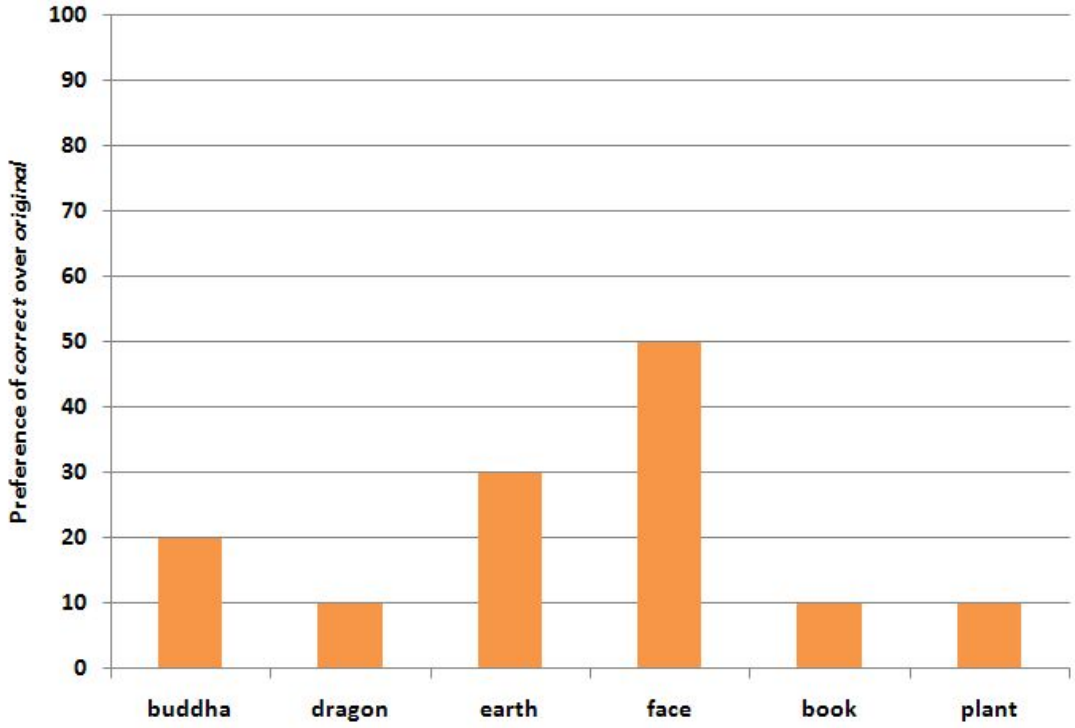


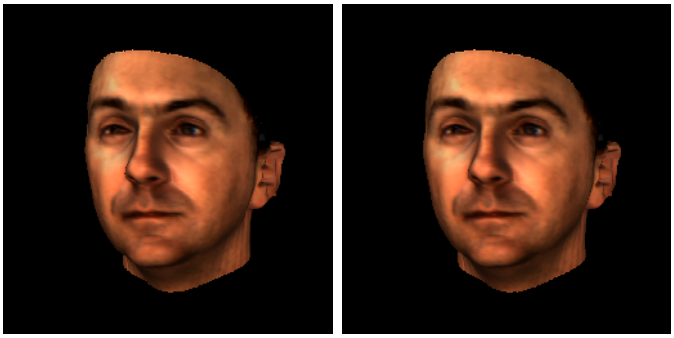
Figure 8.3: Preference of corrected over original condition, broken down by test scene. 50 percent denotes that both conditions are of equal perceived quality.

motivated approaches. First, by focusing on the properties important to visual motion perception, we solve for the better conditioned problem of computing images that are visually convincing, superseding the often ill-posed problem of computing the physically correct interpolation. Second, motion discontinuities are handled perceptually correct by our approach without the need of high-level information such as layers, or figure-ground segmentation.

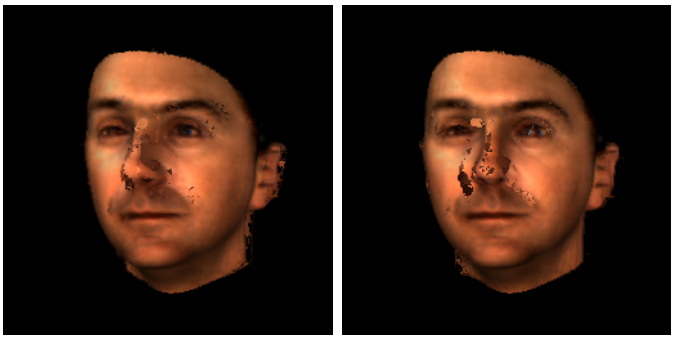
In our user study we validated the overall visual quality of our results and evaluated the contribution of each part of our method. Especially, optimizing motion coherence while correctly handling motion discontinuities significantly improved the perceived quality of the results. In comparison to the rated quality of the original sequences (ground truth) our achieved visual quality is already reasonably close and in addition outperforms the tested other approaches. The finding, that for the face sequence the subjects could not decide if ground truth or our result is better (preference was at 50%) greatly supports the proposed image interpolation approach and shows that "gold-

standard" for some scenes is already achievable.

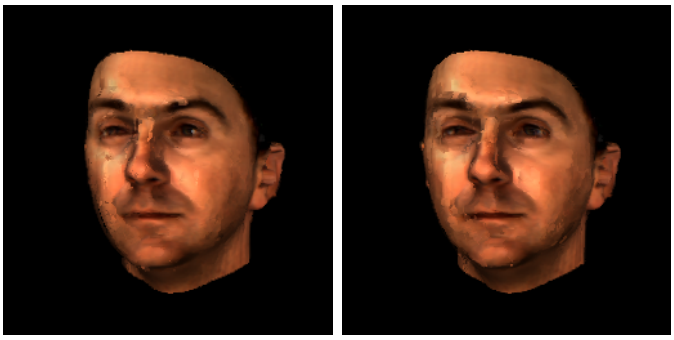
Put into context, the ability to create high-quality image interpolations is beneficial to a wide field of interesting applications: stunning visual effects created with standard cameras, historic movies improved in quality by increasing the frame-rate to modern standards and new possibilities to create stimuli for psychological questionnaires. By looking at concepts of human vision we can identify what is necessary to make the human observer accept the results as physically correct and find new algorithms that are inspired by perception to compute such solutions.



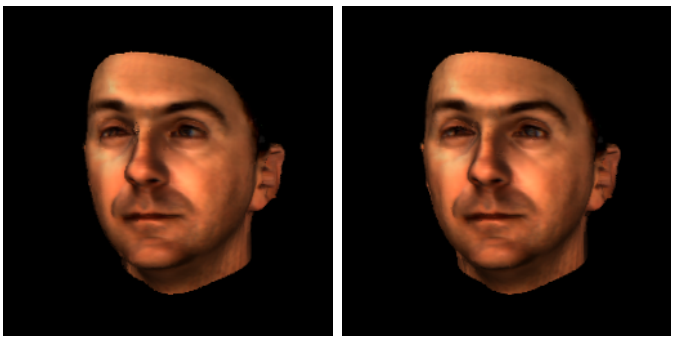
(a) original



(b) opticalflow



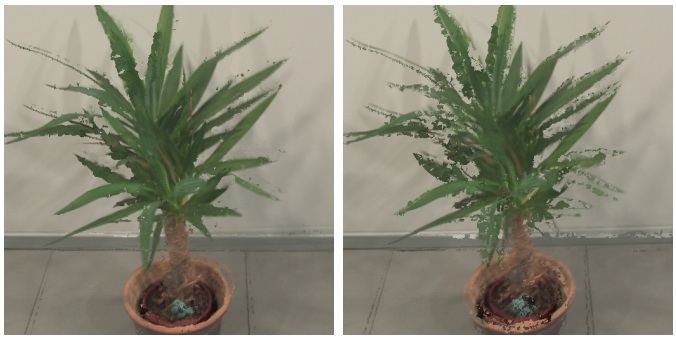
(c) our approach - nooptim



(d) our approach - full

Figure 8.4: Two consecutive images from the face scene sequences. (a) ground truth (b) Horn and Schunck optical flow (c) without optimizing motion coherence (d) automatic interpolation result obtained with our approach.

(a) original



(b) opticalflow

(c) our approach - nooptim

(d) our approach - full

Figure 8.5: Two consecutive images from the plant scene sequences. (a) ground truth (b) Horn and Schunck optical flow (c) without optimizing motion coherence (d) automatic interpolation result obtained with our approach.

8. A PSYCHOPHYSICAL USER-STUDY ON IMAGE INTERPOLATION

9

Estimating Time Difference of Uncalibrated and Non-Stationary Cameras

9.1 Introduction

In this thesis we have introduced two approaches for space-time interpolation of multi-view video data recorded with unsynchronized cameras. In the following two chapters we will discuss how we can make use of such pairwise image interpolation approaches to achieve unconstrained and smooth space-time navigation through unsynchronized and uncalibrated multi-view video data. That is, we work towards the goal to navigate in intuitive directions such as horizontal and vertical view and the time directions alleviating the restriction to only interpolate on paths composed by pairwise image interpolation.

The first step to achieving this is to measure the time difference between the sequences. Specifically, we present a method to find the time offset between two or more recorded video sequences in the general case of unsynchronized, non-stationary cameras up to sub-frame accuracy. We address the problem of identifying the time relation between recorded sequences without the need to intervene in the scene or using special cameras. Our method is based on tracked feature points and the resulting trajectories over time. It is divided into two steps. First we find the time offset up to frame accuracy by extracting salient points of such trajectories and matching their time patterns.

Then using this coarse alignment, we can reformulate the estimation of the fundamental matrix to directly find the time offset of the non-stationary cameras at sub-frame accuracy.

9.2 Problem Formulation

Given two recorded video sequences $S_* = \{I_*^1, I_*^2, \ldots, I_*^k\}$ recorded at the same temporal sampling rate, our goal is to estimate the time difference Δt between S_1 and S_2 such that

$$t_1 + \Delta t = t_2. \tag{9.1}$$

Our goal is thus to estimate Δt, first up to frame accuracy and later up to sub-frame accuracy, from only the recorded image sequences.

The relative camera positions are unknown, and changing over time as the cameras are allowed to move separately. However, to any given time t the epipolar constraints between corresponding points are described by the Fundamental matrix F_t [52]. With $\mathbf{p}_1^{t_1}$ and $\mathbf{p}_2^{t_2}$ the images of the same scene points recorded with the different cameras, we can derive the constraint:

$$(\mathbf{p}_2^{t_1+\Delta t})' \; F_t \; \mathbf{p_1^{t_1}} = 0. \tag{9.2}$$

While single image matching contains no information of the temporal shift, tracking points over multiple frames results in feature trajectories that allow to analyze correspondence on the temporal aspect. We will denote such a trajectory by $T_*^p = \{\mathbf{p}_*^1, \mathbf{p}_*^2, ..., \mathbf{p}_*^k\}$ by tracking the image $\mathbf{p}_*$ of a scene point over k frames. In this paper we assume that trajectories and correspondences between projected scene points are known. They can be obtained using standard algorithms as for example with the Lucas-Kanade-Tomasi feature tracker [78, 127]. Further we expect the camera motion to be smooth and slow compared to the object movement and the recorded scene is supposed to contain linear and non-linear object motion.

9.3 Frame-accurate Temporal Alignment

Our first step in achieving an exact estimation of the temporal offset between two video sequences is to estimate the integer or frame accurate time offset from correspondences

between motion trajectories T_1^p and T_2^p. Since we are recording the same dynamic scene with both cameras, we expect to catch the same movement in both video sequences. But rather than matching the trajectories directly, which is hard to achieve if the cameras are moving independently, we focus only on interest points extracted from the trajectories. Our goal is to find temporal features that represent best the characteristics of the trajectories while achieving maximal view-independence. The frame-accurate temporal misalignment can then be robustly estimated by finding the best alignment between these representations.

The estimation of the temporal offset between cameras of unknown relation and motion requires view-independence of the compared properties. However, the actual values and derivatives of the trajectories are heavily view dependent. Hence a naive approach, e.g. choosing the points depending on time derivatives, is problematic as such values of corresponding trajectories will grow very dissimilar with increasing view angle differences and camera motion. Instead, we rely on the *time-points* of extremal changes of trajectories which are very much view independent, cf. Figure 9.1. Especially, points that build the basis for a linear approximation of a motion trajectory are such points of extremal change. Interestingly, we can resort to an algorithm from a different domain to find these points. The recursive Douglas-Peucker-Algorithm [37] provides a robust simplification of vector lines which is used to scale down coastlines in geographic maps. Applied to trajectories Algorithm 1 results in the points we are interested in, as depicted in Figure 9.1. Here, the motion trajectory of a bouncing ball is reduced to a linear approximation, yielding the interest points at the extremal change positions. Depending on a scale parameter ϵ different degrees of simplification are the result.

Algorithm 1 Douglas-Peucker-Algorithm for extracting motion interest points

Require: Trajectory T, scale parameter ϵ.

1: Connect the first and the last point ($\mathbf{p}^{start}$ and $\mathbf{p}^{end}$) of T with a straight line l.
2: Determine the point $\mathbf{p}^{max}$ of T with the highest distance d to l.
3: If $d > \epsilon$ recursive start the algorithm with the partial trajectories $\{\mathbf{p}^{start}...\mathbf{p}^{max}\}$ and $\{\mathbf{p}^{max}...\mathbf{p}^{end}\}$. Else $\mathbf{p}^{start}$ and $\mathbf{p}^{end}$ are motion interest points.

Since we are interested only in the temporal offset the actual spatial position of the interest points is not of interest. Thus, we can represent the trajectories as binary

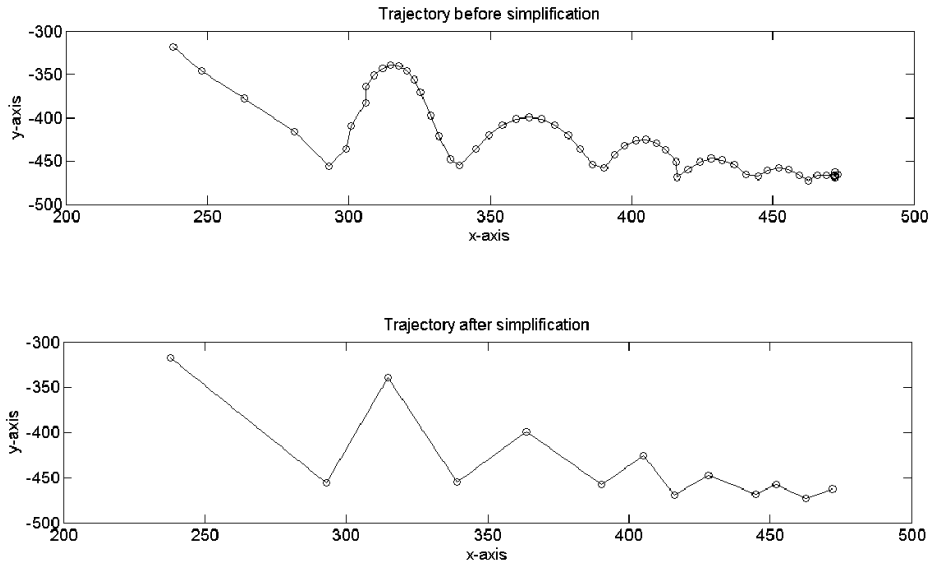


Figure 9.1: 2D motion trajectory of a bouncing ball seen from a single camera. The extraction of interest points on the trajectory can be computed using the Douglas-Peucker-Algorithm.

codes allocating every found motion interest point on the trajectory with a *one* and the remaining frames with a *zero* (cf. Figure 9.2).

Based on this, a one-dimensional, binary representation of the trajectories is computed, and the estimation of the frame accurate time offset can be reposed as the matching of two binary strings. From the many possible algorithms, we implemented our string alignment algorithm based on a simplified Needleman-Wunsch-Algorithm [94] to compute this match (cf. Algorithm 2) robustly and efficiently.

While a single trajectory with sufficient interest points contains enough information to compute the frame accurate solution, we can make use of more than one trajectory. In practice, tracking results and matchings, especially in the case of occlusion, can get unreliable. To compensate for these outliers, we repeat the estimation for each trajectory pair. By counting the results, the offset with the most votes is then chosen as the correct solution.

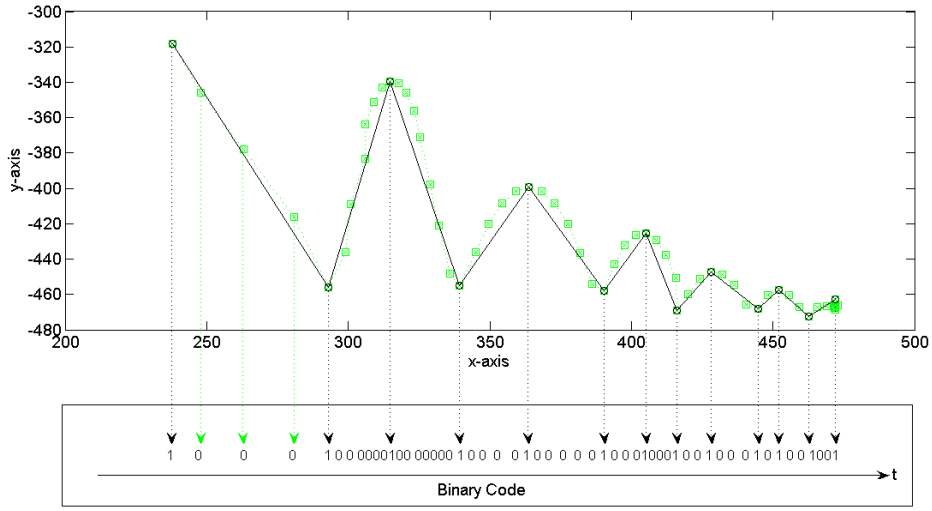


Figure 9.2: Since only the temporal aspect of trajectories are view invariant we introduce a binary code as representation. A *one* represents an extracted interest point and a *zero* any remaining frame of the video sequence.

9.4 Achieving sub-frame accuracy

Using the previously computed frame-accurate temporal offset, corresponding trajectories can already be synchronized up to frame accuracy. But as they are recorded with uncalibrated cameras, there is in general also a sub-frame offset. From now on we assume that T_1^p and T_2^p are in alignment up to frame-accuracy and thus

$$t_2^p = t_1^p + a \tag{9.3}$$

with $0 \leq a < 1$. Approximating the continuous trajectory with linear terms, we further get

$$\mathbf{p}_*^{t+a} \approx (1-a)\, \mathbf{p}_*^t + a\, \mathbf{p}_*^{t+1}. \tag{9.4}$$

Substituting $\mathbf{p}_{t+a}$ for $\mathbf{p}$ in (9.2) yields

$$((1-a)\, \mathbf{p}_2^t + a\, \mathbf{p}_2^{t+1})'\, F_t\, \mathbf{p}_\mathbf{1}^{\mathbf{t_1}} = 0. \tag{9.5}$$

Thus, one pair of corresponding trajectories yields one constraint on the unknown Fundamental matrix F_t. Extracting nine independent motion trajectories from each

Algorithm 2 String Alignment Algorithm

Require: 2 binary codes B, C with $|B| = m$ and $|C| = n$.

1: Construct a $m \cdot n$ table T as follows: $T(i, j) = B(i)\ xor\ C(j)$ (with xor the binary exclusive or).
2: Assign to every diagonal its sum divided by its length as an error measure.
3: Let (i_0, j_0) be the first element of the diagonal with the lowest error value, the time shift between the cameras is given as $i_0 - j_0$ (depending on which camera is ahead, this can be a positive or negative value).
4: As for increasing i_0 (or j_0, one of them has always to be zero) the length of the diagonals is decreasing avoiding accurate results, it is required to assure a minimal length of the diagonals (which means a minimal temporal overlap of the two sequences).

camera sequence allows to generate a system of equations and to estimate the temporal shift directly. At time t, (9.5) can then be reformulated using f_t, the corresponding vector containing the nine unknown entries of F_t in descending order, to

$$M(a)f_t = 0 \tag{9.6}$$

with $M(a)$ a 9×9 matrix in only a single free variable a. As for the correct temporal off-set there must exist a solution to the above equation, the following constraint needs to be satisfied

$$det(M(a)) = 0 \tag{9.7}$$

which corresponds to a degree six polynomial in a (as only the first two coordinates of $\mathbf{p}_t$ are a function of a). For the case of no motion or only camera motion, there is no unique solution for a since the equation holds for all $a \in [0 \dots 1)$. The solution for this will be unique if the trajectories stem from at least two independent motions. This is for example the case for a static background and a moving foreground, two different objects with independent motion or a non-linear deforming object. Hence solving (9.6) for a provides six direct solutions for Δt where at most one solution should lie in the range $[0 \dots 1)$.

In practise, due to noise and tracking errors, the solutions are more robust the more independent the motions of the trajectories are. To measure this independence we first estimate the fundamental matrix from the inter-camera correspondences $\mathbf{p}_1^t$ and $\mathbf{p}_1^{\mathbf{t+1}}$,

F_c [52] and compute the residual error,

$$r_F = \sum [(\mathbf{p}_1^{t+1})' \, F_c \, \mathbf{p}_1^t] \tag{9.8}$$

A large r_F then indicates that the independence assumption is fulfilled, e.g. that the relation does not obey the epipolar constraints between the two time points. Repeating the estimation of the a for different time points, we threshold r_F to compute a solution only if the independence assumption is sufficiently well fulfilled. Analyzing the distribution of the computed time shifts, we expect that the statistical mean gives the true sub-frame offset, cf. Figure 9.3.

9.5 Results

We have applied our synchronization method to various video sequences, both synthetic and real camera data to demonstrate the robustness and applicability of this approach. For generating synthetic data we used 3-dimensional trajectories of tracked markers on moving people provided by the Carnegie Mellon University [27] and then calculated the perspective projection for different camera views. This enabled us to freely specify camera positions and orientations as well as the exact time shift and thus to compare the obtained results with ground truth.

As expected, the robustness of the results depends on the length of the considered trajectory as well as on the baseline of the generated views: a longer trajectory provides more information and thus gives more robust results whereas a wider baseline hinders finding the correct alignment. Figure 9.3 illustrates this relation using only one trajectory for evaluation: in the green area our algorithm for finding the frame-accurate offset provides a correct estimation and fails in the red one. In these test cases we chose ϵ in such a way that no more than fifty and no less than five percent of the points in the trajectory were considered as points of interest. Summarizing the results, one can expect to find the correct solution for image sequences with a length of 100 or more frames for baseline differences up to 45 degrees and even up to 90 degrees for longer trajectories. For the evaluation of the sub-frame accuracy we chose an angle of 20 degrees between the cameras and a time shift of 0.4 frames. We chose 9 correspondences and repeated the estimation of a over a length of 100 frames. The median of

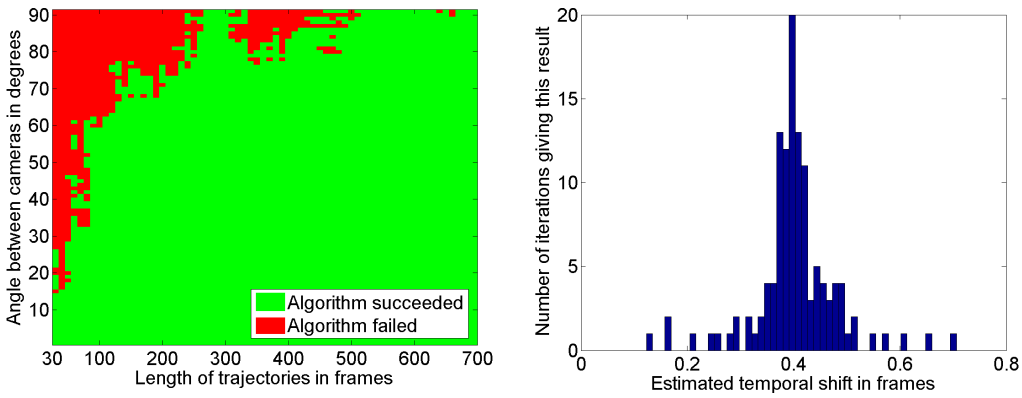


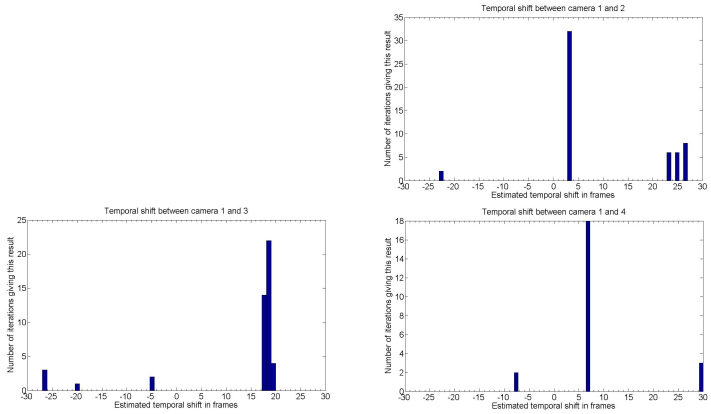
Figure 9.3: Left: the relation between trajectory length and camera baselines. Right: histogram of the estimated possible sub-frame-accurate time shifts with a distinct maximum at 0.40, in accordance with ground truth.

the distribution of possible time shifts was 0.3999 and the mean value was 0.4031 with a variance of 0.0062, both according to ground truth.

In our first experiment with real data we recorded a dancing woman with a set of four stationary, but uncalibrated, cameras. The cameras were placed around the scene next to each other with an angle of 15 degrees to their respective neighbors (so the angle between the first and the last camera were 45 degrees), cf. Figure 9.4(a). The trajectories were obtained by using a pyramid Lucas-Kanade feature tracker [78] with a length of 100 frames each. We only evaluated a pair of two cameras at once and compared the results for different ϵ, effectively changing the amount of extracted time interest points. The results are shown in the histograms in Figure 9.4(b). In Figure 9.4(c) four frames with the correct temporal shift are shown. Computing the offsets for all pairs of cameras, we found that the results are consistent as the sum of all offsets is zero as can be seen in Figure 9.5.

Our second experiment is an outdoor scene with two moving, hand held cameras recording a trial biker. The sequences we used for synchronization were of 100 frames length. We used only one pair of corresponding trajectories. The obtained time shift of 6 frames is according to ground truth, as can be seen in Figure 9.6, and shows that our algorithm is unaffected of the camera movement. For testing the algorithm on sub-frame accuracy and moving cameras, we recorded with one moving hand held and one stationary camera. We chose as scene a single throw of a ball, since here ground

(a) Unsynchronized sequences of a dancing woman from different viewpoints and the overlayed trajectory (blue) of the tracked point (red). The same frame count shows different points in time.



(b) Using different values of ϵ results in a histogram for every pair of cameras.

(c) Using the estimated offsets the videos are correctly synchronized.

Figure 9.4: Frame-accurate temporal alignment of four camera sequences of a dancing woman.

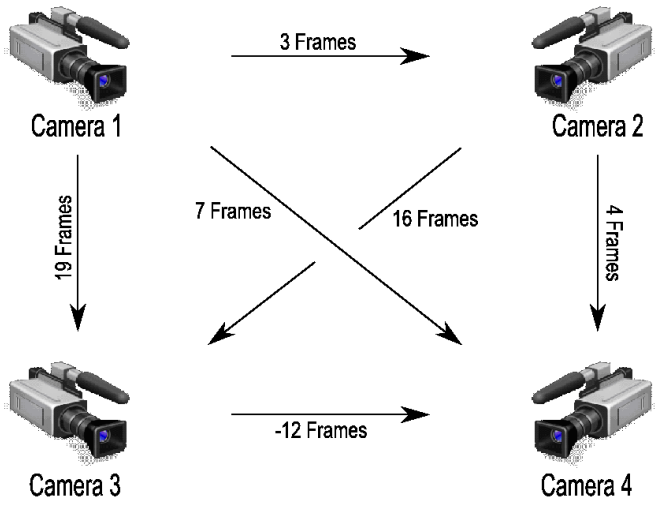


Figure 9.5: Computing the temporal shift for every pair of cameras yields a graph showing the relationships between the cameras. Note that the sum of every loop in this graph must be zero.

truth can at least be roughly estimated due to the simplicity of the object motion. By measuring pixel distances, we manually estimated the time difference to approximately 0.8 frames as ground truth. The motion of the ball provided one trajectory. The remaining eight trajectories were generated by tracking background points. The obtained derivation of possible time shifts had its peak at 0.7681, conforming quite good to the estimated ground truth of 0.8, and a variance of only 0.0019 frames and can be therefore regarded as stable. Also, the variance was even lower than the variance using our synthetic data as the synthetic data contained less linear motion. The trajectories used for this estimation were of 89 frames length but only 10 frames fulfilled the additional robustness constraint as discussed in Section 9.4.

9.6 Summary

We have introduced a method to estimate both the frame accurate and then the sub-frame accurate offset of multi-view video sequences. In contrast to other methods we are able to handle both scene and camera motion without additional information. The first step is based on feature trajectories that are temporally analyzed and reduced to a binary string pattern that is nearly completely view independent. Then the integer off-

set is estimated by robust string matching. With a known integer offset, the sub-frame offset estimation is then based on a reformulation of the fundamental matrix estimation to handle also linear motion of the cameras between recorded frames. We showed the application of our method on real world examples and evaluated the methods for robustness and accuracy on datasets with known ground truth.

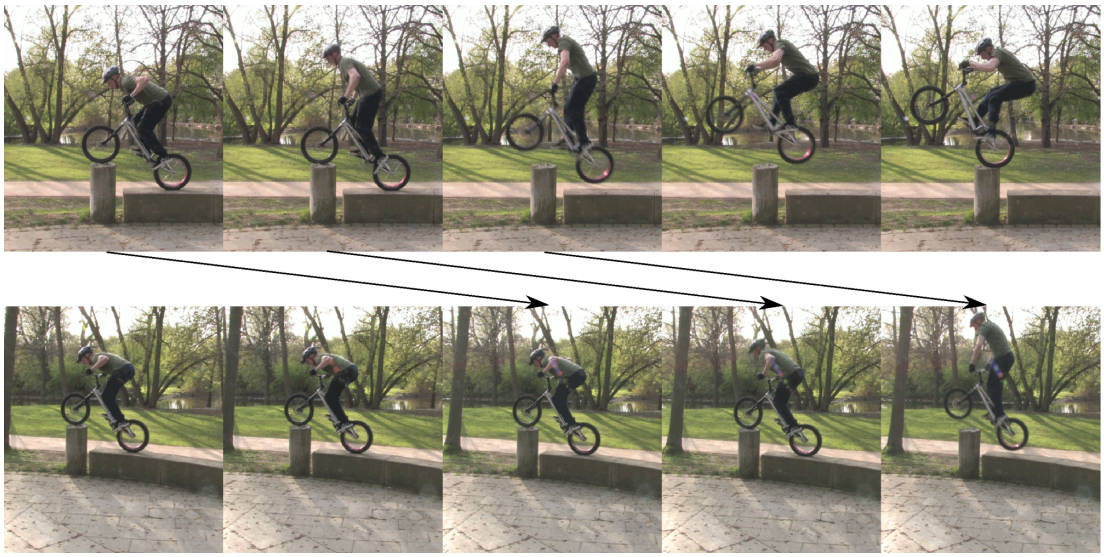


Figure 9.6: Beginning at frame 78, every third image of both video sequences is displayed. The sequences show a time delay of 6 frames.

9. ESTIMATING TIME DIFFERENCE OF UNCALIBRATED AND NON-STATIONARY CAMERAS

10

Multi-View and Time Interpolation in Image Space

10.1 Introduction

In this thesis we have introduced two approaches for space-time interpolation of multi-view video data recorded with unsynchronized cameras. These methods do not rely on an intermediate 3D reconstruction but interpolate directly in image space. One of the greatest advantages is that it becomes possible to interpolate in view and even time from footage captured with standard non-synchronized, uncalibrated cameras. The remaining issue in this image space approach is how virtual cameras describing novel and unrecorded view and time points, can be defined if no 3D information is available.

In this chapter we introduce an embedding of multiple video sequences into a N-D camera-time (typically N=2 or 3) space which solves this issue. In the proposed navigation space each point is one of the images recorded by a camera during acquisition, and multi-video sequences thus form point clouds in this space. By computing the set of simplices that form the delaunay tessellation of these point clouds, relations between the recorded images are defined by edges. These edges then represent pair-wise interpolation between the start-point and end-point image of each edge. A novel point inside the convex hull of the multi-video data in camera-time space can be described by a simplex and barycentric coordinates. This point is then also the description of the virtual camera and we derive a scheme to combine multiple image interpolations to compute the associated in-between image for this specific view and time point.

10.2 Navigation Space

The applicability of image interpolation techniques such as the methods introduced in this thesis and other optical flow estimation approaches [6] are very general because they can be used to render plausible transitions between two images without extra constraints (e.g. epipolar geometry or time synchronization). This allows to create view and/or time interpolation with unsynchronized and uncalibrated cameras. However, views rendered for novel virtual cameras are restricted to lie in-between pairs of recorded images. This is problematic in two ways:

First, movement is not independent. This means if the captured images are not spatially and temporarily aligned, i.e. if unsynchronized and non-uniformly placed cameras are used, each motion between two images will be a composite of several degrees of freedom. Second, movement is not continuous. A global mapping is necessary that defines how more than two images can be interpolated to achieve a smooth navigation through the whole dataset without noticeable transitions between different interpolations.

The main contribution of this chapter is the definition of an N-dimensional space which allows independent and continuous navigation through multi-view video sequences recorded with uncalibrated and unsynchronized cameras. We will refer to this space as navigation space as its axis span independent directions of navigation through the multi-video datasets. A typical navigation space has two or three dimensions depending on the camera setup (cf Figure 10.3). Each image of the multi-view dataset is mapped to a point in this navigation space. The set of all images (from the different cameras) then forms a point cloud. The coordinates of an image in this space can be thought of as the viewing direction of the camera and the instant it was recorded measured in scene time. A similar idea has also been sketched in the seminal work by Chen and Williams [25] for view interpolation. We will discuss its extension to time and also show how image interpolation approaches independent of additional depth information as in [25] is sufficient to create the in-between images. The navigation space also bears some resemblance to the prevailing space-time diagrams used by [161] and [60]. In the following, we introduce how the mapping between images and navigation space coordinates can be computed and how unrecorded points inside the convex hull spanned by the input can be described as a combination of recorded images.

10.2.1 Axis Definition

Figure 10.1: Multi-camera setup for the acquisition of space-time video footage. Up to 16 off the shelf unsynchronized HDV camcorders which recorded to tape on tripods where used.

The navigation space consists of view/space dimensions and a time dimension, representing the navigation directions of the virtual camera. Each image of the recorded sequences is mapped to a unique point in this space. The first mapping we introduce is a mapping of the camera/view positions to navigation space. We denote this embedding the *camera manifold* since we will derive it from the a mapping of the camera positions to a manifold such a a line or a plane. Depending on the camera setup, an embedding that has one or two dimensions is usually sufficient for typical multi-view videos. As stated, the criteria for the mapping is that the axis of navigation space should represent intuitive motion directions such as horizontal and vertical motion and that the distances between cameras are preserved. Figure 10.2 depicts an example of the camera manifold that is used in the camera arrangement shown in Figure 10.1. For the special case of stationary cameras, the embedding of the cameras is constant. A suitable embedding in this typical multi-view recording setup is a cylindrical mapping of the camera positions. This can be recovered quickly by manually marking the image positions of the lens centers in an image of the camera setup, Figure 10.1. A more sophisticated approach is for static cameras to calibrate their positions, e.g. using Zhang's approach [170]. Note however, that in contrast to approaches based on epipolar geometry, the accuracy is not crucial as coarse relations suffice. For non-stationary cameras, the embedding is dynamic and changes in time. In this case camera positions must be tracked over time and projected onto the manifold by more sophisticated approaches like structure from motion [52].

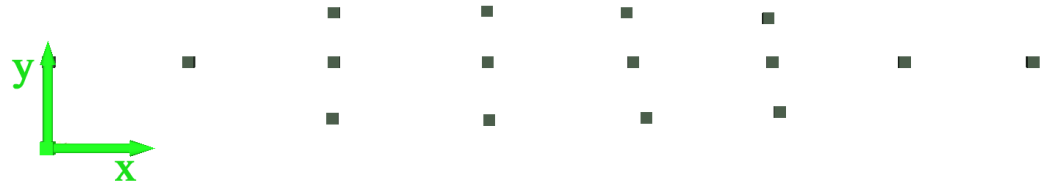


Figure 10.2: Typical camera manifold example. It corresponds with the 16 camera setup in Figure 10.1. Each camera is represented by a cube on the xy-plane.

Since we are dealing with dynamic scenes, one axis of the navigation space is the scene time axis. The time coordinate for each image of a camera is determined by the local camera time of each image and the mapping of the camera time to the global scene time. In contrast to enforcing time synchronizity of the recording cameras, a one to one correspondence between frames of different cameras is not required. Instead it suffices that the time relation is known as for example obtained with the method described in Chapter 9. Thus, time-freeze shots from unsynchronized image sequences are just one-dimensional motions in navigation space.

After the mapping, all recorded images from the multi-video sequence form a point cloud in navigation space, Figure 10.3. The convex hull of this point cloud bounds the space of positions, and thus virtual cameras, that can be rendered with our approach. For the remainder of this chapter we will discuss the 3-dimensional navigation space without restricting the results applicability to lower or higher dimension.

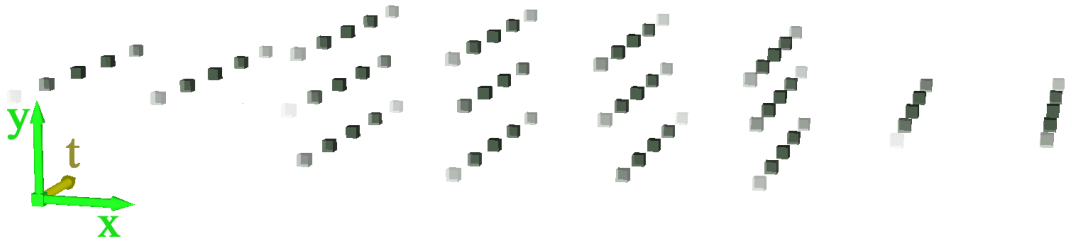


Figure 10.3: The still images from the multi-video sequence form a point cloud in navigation space. Images captured by the same camera are easily identified, as they lie on a straight line that runs across the temporal axis.

10.2.2 Tetrahedralization

To apply an image interpolation approach, the virtual camera must be described as weighted combinations of recorded images. To accomplish this, first a delaunay tetrahedralization of the multi-video point cloud is computed (cf. Figure 10.4). This ensures that each space-time point is inside at least one *space-time tetrahedron*. For each tetrahedron T, the vertices v_1, v_1, v_3, v_4 , referred to as *images vertices*, represent recorded frames from the multi-view sequence. The edges define adjacency between the vertices images. During rendering the dense correspondence fields defined by these edges are used to warp the images.

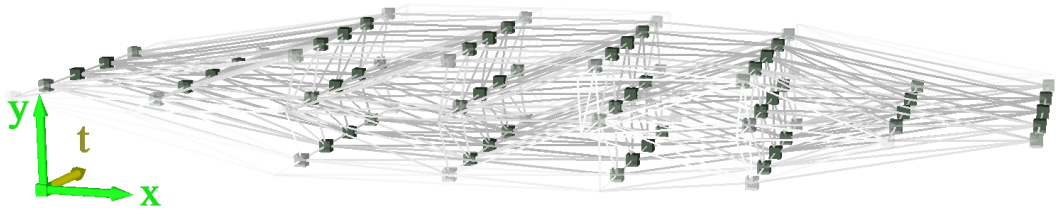


Figure 10.4: Space-time tetrahedra. The delaunay tetrahedralization ensures that each point inside the convex hull of the point cloud lies in one tetrahedron. Edges between vertices images define adjacency between them

Rendering a novel space-time point for a given virtual camera consists of two parts. First we search for a tetrahedron that contains it. Then, the virtual camera is described in terms of vertices images, adjacency between the vertices images and barycentric coordinates.

10.3 Rendering

In this section we describe how a point in navigation space is mapped back to image space. To render novel views corresponding to the virtual camera, I_v the first step is to search for a tetrahedron T that contains p. Then the four images $I_i, i = 1, ..., 4$ and 4 edges of T define a set of images and warpings Wij. Note that all $Wi*$ are defined in the same image basis, and can thus be combined. Then q can be computed by multi-image interpolation as follows: First the combined warping is computed as

$$W_i = w_j W_{ij} + w_k W_{ik} + w_l W_{il} \tag{10.1}$$

where $i, j, k, l = 1, ..., 4$, $i \neq j$, $i \neq k$, $i \neq l$, $j \neq k$, $j \neq l$, $k \neq l$ and the w_* are the barycentric coordinates of p in relation to T. Then, the four images are blended according to their barycentric coordinates.

$$I_v = \sum_{i=1}^{4} (1 - w_i)\ W_i(I_i, w_i) \tag{10.2}$$

Despite a point may lie in more than one tetrahedron at the same time, i.e. if the point p lies exactly on an edge or on one of the corner points we stop our search when we find the first tetrahedron that contains p. In this case, one or more barycentric coordinates are zero and there is no difference, whatever the tetrahedron we use to define p. Figure 10.5 shows an example of our rendering approach.

10.4 Summary

In this chapter we have introduced a representation for unsynchronized and uncalibrated multi-view video data. The goal of this representation is to enable continuous and independent navigation without 3D reconstruction or other additional information. Specifically, we are able to define a virtual camera that enables a user to navigate horizontally, vertically and in time, independent of the recorded video footage. Novel images for virtual cameras are computed by standard image interpolation methods such as the method introduced in the Chapters 6 and 7. The weights and images necessary for the interpolation are derived from a tetrahedralization of the multi-video data. With this method it is possible to achieve image-based real-time interactive and intuitive exploration of the recorded multi-video footage. The benefit is that with our approach, no special synchronization hardware is necessary and independent navigation especially in time, also known as time-freeze shots, are still possible.

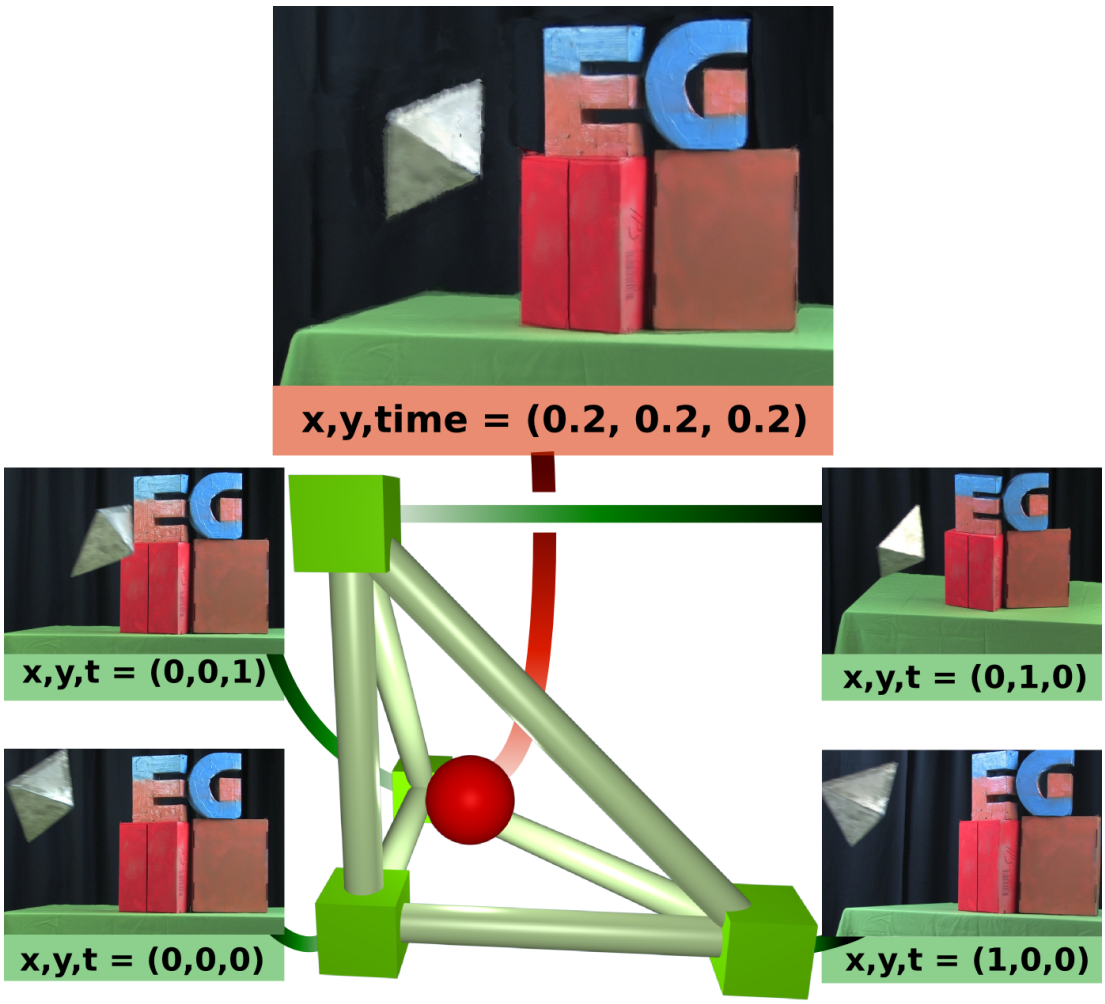


Figure 10.5: Example of view and time interpolation. Four images v_1, v_2, v_3, v_4 (bottom), represented by cubes, are warped and blended into the output image q (top) which is defined by the red ball p in navigation space.

11

Discussion and Conclusions

In the following we summarize our work, discussing contributions and draw-backs of the presented methods. Following that we draw conclusions and present an outlook on future work.

11.1 Summary

First, we introduced an approach for the animation and image space modeling of flames. Our method hereby relies on a statistical learning approach. We propose a general flame model which can be robustly matched to recorded video sequences. By learning the temporal characteristics of the flame shape and appearance, we are able to synthesize arbitrarily long, unique sequences in real-time. Additionally, our flame model can be interactively manipulated to achieve effects like bending the flame while still maintaining the realism of the overall rendered flames. So far we have tested our approach only on a limited number of different flames. The presented approach also assumes a single flame without topology changes, i.e., plumes. Extending the shape model to more chaotic phenomena is, however, a non-trivial task.

Next, we proposed a method for analyzing and synthesizing video sequences of natural phenomena. In contrast to the first method we can handle more chaotic and complex phenomena such as log wood fire. We combined a low-dimensional representation of arbitrary image sequences and an image morphing technique to create realistic in-between images. By interactively segmenting the input into sequences with similar start and end frames, new sequences composed of subsequences can be scripted.

11. DISCUSSION AND CONCLUSIONS

Novel in-between images are computed based on a Monge-Kantorovich derived image morphing method. The method is especially suitable for interpolating image sequences of diffusion processes. As with all methods based on the reordering of images, this method relies on self similarity and quasi periodic phenomena. Also the possibility to interactively control the output is limited as only recorded images can be reordered but no completely new images are created.

Then we focused on general image interpolation methods applicable especially for view and time interpolation. After correspondences for multi-view image sequences have been established we introduced two approaches to improve the image deformation to both reduce the number of necessary correspondences and to improve the visual quality of the transitions. First we optimized the per-feature weights with non-linear optimization and additionally improved the deformation per-pixel based on a standard optical flow method. In combination with our perceptually motivated non-linear blending scheme we are able to render plausible in-between images. For complex scenes and backgrounds, we also used different motion layers, obtained by segmentation of foreground and background. This approach was validated on a set of images taken with standard digital SLR cameras, recording motion at only 4 fps. We computed new in-between images to achieve smooth 30 fps with additional view interpolation. Further, since the linearity assumption for the large motions due to the sparse time sampling is often strongly violated we implemented a non-linear interpolation by decomposing the similarity transformations implied by the matched line features into separate rotation, translation and scaling parameter interpolation. The limitations of this approach is that the method still relies strongly on user interaction. Further, only scenes can be handled that can be easily segmented into separately deformed parts.

Then we introduced a more general and fully automatic approach to image interpolation. Enforcing constraints imposed by projective geometry given two images of similar view and/or time points, we are able to automatically derive dense motion fields between these images. In contrast to other methods, however, we do not rely on the estimation of epipolar geometry constraints and can thus also deal with time differences between recordings. The solution is also perceptually adaptive to achieve more robust results and reduce the ambiguity of the correspondence problem. This adaptive reduction also reduces the computational complexity of the problem. Specifically, we

focus on the motion of edges and ensure coherence of the resulting motion while handling occlusions and motion discontinuities gracefully. We confirm the validity of this approach with a user study that shows that although we are solving only the relaxed problem, users can for some examples not distinguish between ground truth and our results. The study was constructed to validate the connection between the proposed image interpolation algorithm and the perceived quality of the computed results. In practice, we can create plausible in-between images for view differences of up to 20 degrees and motion recorded with 25 fps. While we can handle small and midscale occlusions between the images, large occlusions where complete objects disappear still cause artifacts. This can be partly compensated by user interaction to correct matches.

For measuring the time offset between two, possibly moving, cameras without the need for calibration we developed a non-intrusive method. Based on tracking of corresponding features and analyzing trajectories, we achieve in the first step frame accurate alignment with a single feature and then sub-frame accuracy with 9 feature matches during the process. Finally, we make use of this information and deduce a novel representation of multi-view videos recorded with uncalibrated and unsynchronized camcorders. In our proposed navigation space, the axes correspond to viewing directions, such as horizontal, vertical camera movement, and additionally scene time. By specifying a camera in this novel coordinate system, the user is able to navigate intuitively through the multi-view videos. After Delaunay tessellation of the video point cloud, we can create smooth and independent view and time navigation in real-time by rendering the views of a virtual camera with our approach. We applied our interpolation methods to recordings taken with standard HDV camcorders and are able to show slow-motion, time-freeze and view interpolation results in all combinations. The range, however, is restricted to the convex hull of recordings spanned by the cameras.

11.2 Conclusions

In this dissertation we introduced approaches to image space reconstruction and rendering of dynamic 3D scenes and objects. The focus was on easy acquisition using standard off the shelf hardware without the need to calibrate or time synchronize the cameras. The quality of the achieved results is suitable to create convincing animations of natural phenomena and create smooth view and time interpolations. The produced

results are, however, often not physically correct in the strict sense. Instead of solving exact physical problems, we put the strength and weaknesses of the human visual system to our advantage. The key is to focus on the features that are important for achieving plausible results and to relax the exactness of the result in other regions. Understanding the human observer as part of the solution opens up new possibilities, especially in simplifying complex problems and in finding additional constraints for ill-posed problems. Together, the presented approaches to space-time interpolation can also be used to create special effects even after the acquisition process and allow users to interactively and smoothly navigate through multi-view video footage. Recording with standard cameras aid in reducing the costs and helps bridging the gap between laboratory experiments and real production.

11.3 Future Work

Different lines of research can be pursued to further improve the applicability, performance and quality of the presented methods. The introduced flame model could be extended to handle more complex flame shapes. Another goal would be to model more than one viewing direction, either using 3D reconstruction results or by coupling different billboards for the final rendering.

For the view and time interpolation approaches several directions provide promising research goals. On the reconstruction of the motion it would be favorable to extend the solution to more than just pairs of images. This would further improve robustness and increase the quality especially when large occlusion regions are involved, such as when whole objects get occluded between consecutive frames. Another direction of research would be to also improve the results in the non edge regions. Especially the combination with standard optical flow approaches seems promising. The embedding of the cameras in navigation space has only been applied to static cameras so far. However, it is possible to also handle dynamic camera setups. This would involve tracking the relative camera positions over time. While the rendering stays still the same for dynamic setups, an interesting problem is how to intuitively navigate if the recording cameras are not fixed and thus the virtual camera is bound to this underlying motion.

Bibliography

[1] Adelson, E., and Bergen, J. The Plenoptic Funcion and the Elements of Early Vision. In *Computational Models of Visual Processing*, M. Landy and J. Movshon, Eds. MIT Press, 1991, pp. 3–20.

[2] Adelson, E., and Movshon, A. The Perception of coherent Motion in two-dimensional Patterns. In *ACM Siggraph and Sigart Interdisciplinary Workshop on Motion: Representation and Perception* (1983), pp. 11–16.

[3] Alexa, M., Cohen-Or, D., and Levin, D. As-rigid-as-possible shape interpolation. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2000), pp. 157–164.

[4] Atcheson, B., Ihrke, I., Heidrich, W., Tevs, A., Bradley, D., Magnor, M., and Seidel, H.-P. Time-resolved 3D Capture of Non-stationary Gas Flows. *Proceedings ACM Asia Conference on Computer Graphics and Interactive Techniques (SIGGRAPH Asia 2008) 27*, 5 (2008), article 132.

[5] Baker, S., and Matthews, I. Lucas-Kanade 20 Years On: A unifying framework. *International Journal of Computer Vision 56*, 3 (2004), 221–255.

[6] Baker, S., Scharstein, D., Lewis, J., th, S. R., Black, M., and Szeliski, R. A Database and Evaluation Methodology for Optical Flow. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (2007), pp. 1–8.

[7] Barron, J., Fleet, D., and Beauchemin, S. Performance of Optical Flow Techniques. *International Journal of Computer Vision 12*, 1 (1994), 43–77.

[8] Beaudoin, P. Realistic and Controllable Fire Simulation. In *Proceedings Conference on Graphics Interface* (2001), pp. 159–166.

[9] Beier, T., and Neely, S. Feature-Based Image Metamorphosis. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1992), pp. 35–42.

[10] Belongie, S., Malik, J., and Puzicha, J. Matching Shapes. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (2001), pp. 454 – 461.

[11] Bertsekas, D. Auction Algorithms for Network Flow Problems: A tutorial Introduction. *Computational Optimization and Applications 1* (1992), 7–66.

[12] Bhat, K., Seitz, S., Hodgins, J., and Khosla, P. Flow-based Video Synthesis and Editing. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2004), pp. 360–363.

[13] Bichsel, M. Optimizing image morph fields for automatic interpolation of face images. Tech. Rep. 96.03, University of Zurich, 1996.

[14] Black, M. J., and Anandan, P. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding 63*, 1 (1996), 75–104.

[15] Blake, A., North, B., and Isard, M. Learning Multi-Class Dynamics. In *Advances in Neural Information Processing Systems* (1999), pp. 389–395.

[16] Blanz, V., Basso, C., Vetter, T., and Poggio, T. Reanimating Faces in Images and Video. In *Proceedings European Conference on Computer Graphics (EG)* (2003), pp. 641–650.

[17] Boykov, Y., and Funka-Lea, G. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision 40*, 2 (2006), 109–131.

[18] Boykov, Y., and Kolmogorov, V. The MAXFLOW algorithm. http://www.cs.cornell.edu/People/vnk/software.html.

[19] Bregler, C., Covell, M., and Slaney, M. Video Rewrite: Driving Visual Speech with Audio. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1997), pp. 353–360.

[20] Buehler, C., Bosse, M., McMillan, L., Gortler, S., and Cohen, M. Unstructured Lumigraph Rendering. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2001), pp. 425–432.

[21] Canny, J. A Computational Approach To Edge Detection. *Transactions on Pattern Analysis and Machine Intelligence 8* (1986), 679–714.

[22] Carranza, J., Theobalt, C., Magnor, M., and Seidel, H. P. Free-Viewpoint Video of Human Actors. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2003), pp. 569–577.

[23] Caspi, Y., Simakov, D., and Irani, M. Feature-based sequence-to-sequence matching. *International Journal of Computer Vision 68*, 1 (2006), 53–64.

[24] Chartrand, R., Vixie, K., Wohlberg, B., and Bollt, E. A gradient descent solution to the Monge-Kantorovich problem. Preprint: LA-UR-04-6305, 2005.

[25] Chen, S., and Williams, L. View Interpolation for Image Synthesis. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1993), pp. 279–288.

[26] CIE. Colorimetry, publication no.15, supplement no. 2, 1976.

[27] CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/.

[28] Dai, C., Zheng, Y., and Li, X. Subframe video synchronization via 3d phase correlation. In *Proceedings IEEE International Conference on Image Processing (ICIP)* (2006), pp. 501–504.

[29] de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.-P., and Thrun, S. Performance capture from sparse multi-view video. *ACM Transactions on Graphics 27*, 3 (2008), 1–10.

[30] Debevec, P., Borshukov, G., and Yu, Y. "efficient view-dependent image-based rendering with projective texture-mapping". In *Proceedings Eurographics Rendering Workshop (EGRW)* (1998), pp. 105–116.

[31] Dellaert, F., Seitz, S., Thorpe, C., and Thrun, S. EM, MCMC, and Chain Flipping for Structure from Motion with Unknown Correspondence. *Machine Learning 50*, 1-2 (2003), 45–71.

[32] Dempster, A., Laird, N., and Rubinet, D. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society B 39*, 1 (1977), 1–38.

[33] Ditchburn, R., and Ginsborg, B. Vision with a Stabilized Retinal Image. *Nature 170* (1952), 36–37.

[34] Domke, J., and Aloimonos, Y. A Probabilistic Notion of Correspondence and the Epipolar Constraint. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2006), pp. 1–8.

[35] Doretto, G., Chiuso, A., Wu, Y. N., and Soatto, S. Dynamic textures. *International Journal of Computer Vision 51*, 2 (2003), 91–109.

[36] Doretto, G., and Soatto, S. Editable dynamic textures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Madison, Wisconsin, USA, June 2003), vol. 2, pp. 137–142.

[37] Douglas, D., and Peucker, T. Algorithms for the reduction of the number of points required to represent a line or its caricature. *The Canadian Cartographer 10*, 2 (1973), 112–122.

[38] Eisemann, M., Decker, B. D., Magnor, M., Bekaert, P., de Aguiar, E., Ahmed, N., Theobalt, C., and Sellent, A. Floating Textures. *Proceedings European Conference on Computer Graphics (EG) 27*, 2 (4 2008), 409–418.

[39] Ezzat, T., Geiger, G., and Poggio, T. Trainable Videorealistic Speech Animation. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2002), pp. 388–398.

[40] Ezzat, T., and Poggio, T. Visual Speech Synthesis by Morphing Visemes. *International Journal of Computer Vision 38*, 1 (2000), 45–57.

[41] Fedkiw, R., Stam, J., and Jenssen, H. W. Visual Simulation of Smoke. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2001), pp. 15–22.

[42] Felzenszwalb, P., and Huttenlocher, D. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision 59* (2004), 167–181.

[43] Foster, N., and Metaxas, D. Realistic animation of liquids. *Graphical Models and Image Processing 85*, 5 (1996), 471–485.

[44] Foster, N., and Metaxas, D. Modeling the motion of a hot, turbulent gas. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1997), pp. 181–188.

[45] Gelb, A., Ed. *Applied optimal Estimation*. MIT Press, 1974.

[46] Glasbey, C., and Mardia, K. A Review of Image Warping Methods. *Journal of Applied Statistics 25* (1998), 155–171.

[47] Glasbey, C., and Mardia, K. A Penalised Likelihood Approach to Image Warping. *Journal of the Royal Stastics Society B 63* (2001), 462–492.

[48] Goldluecke, B., and Magnor, M. Space-Time Isosurface Evolution for Temporally Coherent 3D Reconstruction. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Washington, D.C., USA, July 2004), vol. I, IEEE Computer Society, IEEE Computer Society, pp. 350–355.

[49] Goldluecke, B., and Magnor, M. Weighted Minimal Hypersurfaces and Their Applications in Computer Vision. In *Proceedings European Conference on Computer Vision (ECCV)* (Prague, Czech Republic, May 2004), vol. 3022 of *Lecture Notes in Computer Science*, Springer, pp. 366–378.

[50] Gortler, S., Grzeszczuk, R., Szeliski, R., and Cohen, M. The Lumigraph. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1996), pp. 43–54.

[51] Haker, S., Zhu, L., Tannenbaum, A., and Angenent, S. Optimal Mass Transport for Registration and Warping. *International Journal of Computer Vision 60*, 3 (2004), 225–240.

[52] Hartley, R., and Zisserman, H. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[53] Hasinoff, S., and Kutulakos, K. Photo-Consistent 3D Fire by Flame-Sheet Decomposition. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (2003), pp. 1184–1191.

[54] Heeger, D., Boynton, G., Demb, J., Seidemann, E., and Newsome, W. Motion opponency in visual cortex. *J. Neurosci. 19* (Aug 1999), 7162–7174.

[55] Horn, B., and Schunck, B. Determining Optical Flow. *Artificial Intelligence 17* (1981), 185–203.

[56] Hubel, D. *Eye, Brain, and Vision*, 2nd ed. W. H. Freeman, 1995.

[57] Hubel, D., and Wiesel, T. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 195 (1962), 106–154.

[58] Ihrke, I., and Magnor, M. Image-Based Tomographic Reconstruction of Flames. In *SCA* (June 2004), pp. 367–375.

[59] Ihrke, I., and Magnor, M. Adaptice grid optical tomography. *Graphical Models 68*, 5 (2006), 484–495.

[60] Inc., D. A. Digital air techniques. http://www.digitalair.com/techniques/index.html, 2007.

[61] Isard, M., and Blake, A. Contour Tracking by Stochastic Propagation of Conditional Density. In *Proceedings European Conference on Computer Vision (ECCV)* (1996), pp. 343–356.

[62] Julesz, B. Textons, the elements of texture perception, and their interactions. *Nature*, 290 (1981), 91–97.

[63] Klein, F. *Vergleichende Betrachtungen über neuere geometrische Forschungen*. Erlangen: A. Deichert, 1872.

[64] Kumar, M. P., Torr, P. H. S., and Zisserman, A. Learning Layered Motion Segmentation of Video. *International Journal of Computer Vision 76* (2005), 301–319.

[65] Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2003), pp. 277–286.

[66] Lamorlette, A., and Foster, N. Structural Modeling of Flames for a Production Environment. *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2002), 729–735.

[67] Land, E., and McCann, J. Lightness and Retinex Theory. *Journal of the Optical Society of America 61* (1971), 1–11.

[68] Leordeanu, M., and Hebert, M. A Spectral Technique for Correspondence Problems using Pairwise Constraints. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (October 2005), vol. 2, pp. 1482 – 1489.

[69] Lerios, A., Garfinkle, C. D., and Levoy, M. Feature-based Volume Metamorphosis. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1995), pp. 449–456.

[70] Levoy, M., and Hanrahan, P. Light Field Rendering. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1996), pp. 31–42.

[71] Lhuillier, M., and Quan, L. Image Interpolation by Joint View Triangulation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1999), pp. 139–145.

[72] Lhuillier, M., and Quan, L. Match Propagation for Image-Based Modeling and Rendering. *Transactions on Pattern Analysis and Machine Intelligence 24*, 8 (2002), 1140–1146.

[73] Liu, C., Torralba, A., Freeman, W. T., Durand, F., and Adelson, E. H. Motion Magnification. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2005), pp. 519–526.

[74] Livingstone, M., and Hubel, D. Anatomy and physiology of a color system in the primate visual cortex. *Jornal of Neuroscience*, 4 (1984), 309–356.

[75] Ljung, L. *System Identification: Theory for the User*. Prentice Hall, 1999.

[76] Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[77] Lucas, B., and Kanade, T. "an iterative image registration technique with an application to stereo vision". In *Proceedings of the International Joint Conference on Artificial Intelligence* (1981), pp. 674–679.

[78] Lucas, B., and Kanade, T. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging understanding workshop* (1981), pp. 121–130.

[79] Manning, R., and Dyer, C. Interpolating view and scene motion by dynamic view morphing. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 1* (1999), –394 Vol. 1.

[80] Mark, W., McMillan, L., and Bishop, G. Post-Rendering 3D Warping. In *Proceedings of the Symposium on Interactive 3D Graphics* (1997), pp. 7–16.

[81] Marr, D. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1983.

[82] Marr, D., and Hildreth, E. Theory of Edge Detection. *Proc. of the Royal Society of London. Series B, Biological Sciences 207* (1980), 187–217.

[83] Marr, D., and Ullman, S. Directional selectivity and its use in early visual processing. *Proceedings of the Royal Society of London, Series B 211* (1981), 151–180.

[84] Matusik, W., Buehler, C., Raskar, R., Gortler, S., and McMillan, L. Image-Based Visual Hulls. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2000), pp. 369–374.

[85] McMillan, L., and Bishop, G. Plenoptic Modeling: An Image-Based Rendering System. *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1995), 39–46.

[86] McNamara, A., Treuille, A., Popović, Z., and Stam, J. Fluid control using the adjoint method. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New York, NY, USA, 2004), ACM, pp. 449–456.

[87] Meyer, B., Stich, T., Magnor, M., and Pollefeys, M. Subframe Temporal Alignment of Non-Stationary Cameras. In *Proceedings British Machine Vision Conference* (2008), pp. 103–112.

[88] Mikolajczyk, K., and Schmid, C. A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence 27*, 10 (Oct. 2005), 1615–1630.

[89] Monge, G. Mémoire sur la théorie des déblais at des remblais. *Histoire de l'Académie Royale des Sciences de Paris* (1781), 666–704.

[90] Mori, G., Belongie, S., and Malik, J. Efficient Shape Matching Using Shape Contexts. *Transactions on Pattern Analysis and Machine Intelligence 27*, 11 (2005), 1832–1837.

[91] Morrone, M., Burr, D., and Vaina, L. Two stages of Visual Processing for Radial and Temporal Motion. *Nature 376* (1995), 507–509.

[92] Movshon, J., Adelson, E., Gizzi, M., and Newsome, W. The analysis of moving visual patterns. In *Study Group on Pattern Recognition Mechanisms*, C. Chagas, R. Gatass, and C. Gross, Eds. Pontificia Academia Scientiarium, 1985.

[93] Mukundan, R., and Ramakrishnan, K. R. *Moment Functions in Image Analysis*. World Scientific, 1998.

[94] Needleman, S., and Wunsch, C. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology 48* (1970), 443–453.

[95] Newton, I. *Opticks or A Treatise of the Reflections, Refractions, Inflections and Colours of Light*. Dover Publications, 1952.

[96] Nguyen, D. Q., Fedkiw, R., and Jensen, H. W. Physically Based Modeling and Animation of Fire. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2002), pp. 721 – 728.

[97] North, B., and Blake, A. Learning and Classification of Complex Dynamics. *Transactions on Pattern Analysis and Machine Intelligence 22*, 9 (2000), 1016–1034.

[98] O'Sullivan, C., Howlett., S., Mcdonnell, R., Y.Morvan, and K. O'Conor, K. Perceptually Adaptive Graphics. *In Eurographics, State-of-the-art-Report 6* (2004).

[99] Oyster, C. *The Human Eye: Structure and Function*. Sinauer Associates, 1999.

[100] Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. Highly accurate Optic Flow Computation with Theoretically Justified Warping. *International Journal of Computer Vision 67*, 2 (2006), 141–158.

[101] Pattanaik, S. N., Ferwerda, J. A., Fairchild, M. D., and Greenberg, D. P. A multiscale model of adaptation and spatial vision for realistic image display. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1998), pp. 287–298.

[102] Perona, P., and Malik, J. Scale-Space and Edge Detection using Anisotropic Diffusion. *Transactions on Pattern Analysis and Machine Intelligence 12*, 7 (1990), 629–639.

[103] Piccardi, M. Background Subtraction Techniques: A Review. pp. 3099 – 3104.

[104] Pighin, F., Szeliski, R., and Salesin, D. Modeling and Animating Realistic Faces from Images. *International Journal of Computer Vision 50*, 2 (2004), 143–169.

[105] Pless, R. Image Spaces and Video Trajectories: Using Isomap to Explore Video Sequences. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (2003), vol. 2, pp. 1433–1441.

[106] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[107] Qian, N., and Andersen, R. A physiological model for motion-stereo integration and a unified explanation of Pulfrich-like phenomena. *Vision Res. 37* (Jun 1997), 1683–1698.

[108] Ramanarayanan, G., Bala, K., and Ferwerda, J. Perception of Complex Aggregates. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2008), pp. 1–10.

[109] Ramanarayanan, G., Ferwerda, J., Walter, B., and Bala, K. Visual Equivalence: Towards a New Standard for Image Fidelity. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2007), pp. 654–663.

[110] Reeves, W. PARTICLE Systems - A technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics 2* (1983), 91–108.

[111] Reichardt, W. Autocorrelation, A principle for the evaluation of sensory information by the central nervous system. In *Sensory communication*, W. Rosenblith, Ed. New York: MIT Press-Willey, 1961, p. 303–317.

[112] Reinhard, E., and Devlin, K. Dynamic Range Reduction Inspired by Photoreceptor Physiology. *IEEE Transactions on Visualization and Computer Graphics 11*, 1 (2005), 13–24.

[113] Reinhard, E., Ward, G., Pattanaik, S., and Debevec, P. *High Dynamic Range Imaginging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann Series in Computer Graphics, 2005.

[114] Ren, X., and Malik, J. Learning a classification model for segmentation. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (2003), pp. 10– 17.

[115] Riggs, L., and Ratliff, F. The effects of counteracting the normal movements of the eye. *Journal of the Optical Society of America 42* (1952), 872–873.

[116] Rubner, Y., Tomasi, C., and Guibas, L. A Metric for Distributions with Applications to Image Databases. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (1998), pp. 59–66.

[117] Ruzon, M., and Tomasi, C. Color Edge Detection with the Compass Operator. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1999), pp. 160–166.

[118] Sand, P., and Teller, S. Particle Video: Long-Range Motion Estimation using Point Trajectories. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 2195–2202.

[119] Schaefer, S., McPhail, T., and Warren, J. Image Deformation Using Moving Least Squares. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2006), pp. 533–540.

[120] Schirrmacher, H., Heidrich, W., and Seidel, H.-P. High-quality interactive lumigraph rendering through image warping. In *Proceedings Graphics Interface* (2000), pp. 87–94.

[121] Schirrmacher, H., Li, M., and Seidel, H.-P. On-the-Fly Processing of Generalized Lumigraphs. *Coputer Graphics Forum 20*, 3 (2001), 165–174.

[122] Schödl, A., and Essa, I. A. Controlled Animation of Video Sprites. In *SCA* (New York, NY, USA, 2002), ACM, pp. 121–127.

[123] Schoedl, A., Szeliski, R., Salesin, D., and Essa, I. Video Textures. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2000), pp. 489–498.

[124] Schoenemann, T., and Cremers, D. High Resolution Motion Layer Decomposition using Dual-space Graph Cuts. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Anchorage, Alaska, June 2008).

[125] Seitz, S., and Dyer, C. Physically-Valid View Synthesis by Image Interpolation. In *Proceedings Workshop on Representation of Visual Scenes* (1995), pp. 18–25.

[126] Seitz, S., and Dyer, C. View Morphing. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1996), pp. 21–30.

[127] Shi, J., and Tomasi, C. Good features to track. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Jun 1994), 593–600.

[128] Shum, H.-Y., and Szeliski, R. Construction and refinement of panoramic mosaics with global and local alignment. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (Washington, DC, USA, 1998), IEEE Computer Society, p. 953.

[129] Snavely, N., Seitz, S., and Szeliski, R. Photo Tourism: Exploring Photo Collections in 3D. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2006), pp. 835–846.

[130] Soatto, S., Doretto, G., and Wu, Y. N. Dynamic Textures. *International Journal of Computer Vision 51*, 2 (2003), 91–109.

[131] Spencer, L., and Shah, M. Temporal synchronization from camera motion. In *Proceedings of Asian Conference on Computer Vision* (2004), pp. 515–520.

[132] Stam, J. Stable Fluids. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1999), pp. 121–128.

[133] Stam, J., and Fiume, E. Depiction of Fire and Other Gaseous Phenomena Using Diffusion Processes. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1995), pp. 129–136.

[134] Starck, J., and Hilton, A. Surface capture for performance based animation. *IEEE Computer Graphics and Applications 27(3)* (2007), 21–31.

[135] Stark, M., and Schiele, B. How Good are Local Features for Classes of Geometric Objects. *Proceedings IEEE International Conference on Computer Vision (ICCV)* (Oct. 2007), 1–8.

[136] Stich, T., Linz, C., Albuquerque, G., and Magnor, M. View and Time Interpolation in Image Space. *Computer Graphics Forum (Proceedings of the Pacific Graphics Conference)* (2008).

[137] Stich, T., Linz, C., Wallraven, C., Cunningham, D., and Magnor, M. Perception-motivated Interpolation of Image Sequences. In *Proceedings ACM Symposium on Applied Perception in Graphics and Visualization (APGV)* (2008), pp. 97–106.

[138] Stich, T., and Magnor, M. Learning Flames. In *Proceedings of Vision, Modeling and Visualization Conference (VMV)* (2005), pp. 65–70.

[139] Stich, T., and Magnor, M. Keyframe Animation from Video. In *Proceedings IEEE International Conference on Image Processing (ICIP)* (2006), pp. 2713–2716.

[140] Stich, T., and Magnor, M. Image morphing for space-time interpolation. In *Technical Report / Computer Graphics Lab, TU Braunschweig; 2007-4-2* (2007).

[141] Stich, T., and Magnor, M. Image Morphing for Space-Time Interpolation. In *SIGGRAPH sketches* (2007).

[142] Stich, T., Tevs, A., and Magnor, M. Global Depth from Epipolar Volumes - A General Framework for Reconstructing Non-Lambertian Surfaces. In *Proceedings International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2006), pp. 1–8.

[143] Szewczyk, R., Ferencz, A., Andrews, H., and Smith, B. C. Motion and feature-based video metamorphosis. In *Proceedings of the ACM International Conference on Multimedia* (1997), pp. 273–281.

[144] Tenenbaum, J., de Silva, V., and Langford, J. A global geometric Framework for nonlinear dimensionality reduction. *Science 290* (2000), 2319–2323.

[145] Treuille, A., McNamara, A., Popović, Z., and Stam, J. Keyframe control of smoke simulations. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New York, NY, USA, 2003), ACM, pp. 716–723.

[146] Triggs, B. Joint Feature Distributions for Image Correspondence. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (2001), pp. 201–208.

[147] Turk, M., and Pentland, A. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience 3*, 1 (1991), 71–86.

[148] Vangorp, P., and Dutré, P. Shape-dependent gloss correction. In *Proceedings ACM Symposium on Applied Perception in Graphics and Visualization (APGV)* (2008), pp. 123–130.

[149] Vangorp, P., Laurijssen, J., and Dutré, P. The Influence of Shape on the Perception of Material Reflectance. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2007), pp. 1–9.

[150] Vedula, S., Baker, S., and Kanade, T. Image Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events. *ACM Transactions on Graphics 24*, 2 (2005), 240–261.

[151] Wallach, H. Ueber visuell wahrgenommene Bewegungsrichtung. *Psychologische Forschung 20* (1935), 325–380.

[152] Wallraven, C., Bülthoff, H. H., Fischer, J., Cunningham, D. W., and Bartz, D. The Evaluation of Real-World and Computer-Generated Stylized Facial Expressions. *ACM Transactions on Applied Perception 4*, 3 (2007), 1–24.

[153] Wang, H., and Yang, R. Towards space: time light field rendering. In *Proceedings of thesymposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), ACM, pp. 125–132.

[154] Wang, J., and Adelson, E. Layered Representation for Motion Analysis. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1993), pp. 361–366.

[155] Weng, Y., Shi, X., Bao, H., and Zhang, J. Sketching MLS image deformations on the GPU. *Computer Graphics Forum 27*, 7 (2008), 1789–1796.

[156] Wertheimer, M. Laws of organization in perceptual forms. In *A Source Book of Gestalt Psychology*, W. Ellis, Ed. Kegan Paul, Trench, Trubner & Co. Ltd., 1938, pp. 71–88.

[157] Whitehead, A., Laganiere, R., and Bose, P. Temporal synchronization of video sequences in theory and in practice. *Motion and Video Computing 2* (2005), 132–137.

[158] Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E., Barth, A., Adams, A., Horowitz, M., and Levoy, M. High performance imaging using large camera arrays. *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2005), 765–776.

[159] Wilburn, B., Smulski, M., Lee, H.-H. K., and Horowitz, M. The Light Field Video Camera. In *Proceedings of Media Processors* (2002), pp. 1–8.

[160] Wolberg, G. Image Warping: A Survey. *Visual Computer 14* (1998), 360–372.

[161] Wolf, M. "space, time, frame, cinema: Exploring the possibilities of spatiotemporal effects". *New Review of Film and Television Studies* (Dec. 2006), 369–374. www.digitalair.com/techniques/STFC.pdf.

[162] Xiao, J., Rao, C., and Shah, M. View Interpolation for Dynamic Scenes. In *Proceedings European Conference on Computer Graphics (EG)* (2002), pp. 153–162.

[163] Xiao, J., and Shah, M. Tri-view morphing. *Computer Vision and Image Understanding 96* (2004), 345–366.

[164] Xu, L., Chen, J., and Jia, J. Segmentation Based Variational Model for Accurate Optical Flow Estimation. In *Proceedings European Conference on Computer Vision (ECCV)* (2008).

[165] Yan, J., and Pollefeys, M. Video Synchronization via Space-Time Interest Point Distribution. In *Advanced Concepts for Intelligent Vision Systems* (2004), pp. 501–504.

[166] Yarbus, A. *Eye Movements and Vision*. New York: Plenum Press, 1967.

[167] Young, T. On the theory of light and colors. *Philosophical Transactions of the Royal Society 91* (1802), 12–49.

[168] Yuille, A. L., and Poggio, T. A. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analyis and Machine Intelligence (PAMI) 8*, 1 (1986), 15–25.

[169] Zanella, V., and Fuentes, O. An Approach to Automatic Morphing of Face Images in Frontal View. In *Proceedings Mexican International Conference on Artificial Intelligence* (2004), pp. 679–687.

[170] Zhang, Z. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings IEEE International Conference on Computer Vision (ICCV)* (1999), pp. 666–673.

[171] Zhu, S.-C., Guo, C.-E., Wang, Y., and Xu, Z. What are textons? *International Journal of Computer Vision 62*, 1-2 (2005), 121–143.

[172] Zitnick, C., Jojic, N., and Kang, S. B. Consistent segmentation for optical flow estimation. *Proceedings IEEE International Conference on Computer Vision (ICCV) 2* (Oct. 2005), 1308–1315 Vol. 2.

[173] Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., and Szeliski, R. High-Quality Video View Interpolation Using a Layered Representation. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2004), pp. 600–608.

Curriculum Vitæ - Lebenslauf

Curriculum Vitæ

1978	born in Miltenberg am Main, Germany
1997	Highschool degree, main subjects physics and mathematics
	Gymnasium Neuenbürg, Germany
1998 - 2004	Diploma in Computer Science
	Universität Mannheim, Germany
2005 - 2006	Ph.D. Student in Computer Science, Prof. M. Magnor
	Max-Planck-Institut für Informatik, Saarbrücken, Germany
since 2006	Ph.D. Student Computer Science, Prof. M. Magnor
	TU Braunschweig, Germany

Lebenslauf

1978	geboren in Miltenberg am Main
1997	Allgemeine Hochschulreife
	Gymnasium Neuenbürg
1998 - 2004	Diplom technische Informatik
	Universität Mannheim
2005 - 2006	Wissenschaftlicher Mitarbeiter, Prof. M. Magnor
	Max-Planck-Institut für Informatik, Saarbrücken
since 2006	Wissenschaftlicher Mitarbeiter, Prof. M. Magnor
	TU Braunschweig